

# HP 35s 공학용 계산기

---

## 사용 설명서



제1판

HP 부품 번호 F2215AA-90016

---

## 고지 사항

다음 사이트에서 제품을 등록하십시오. [www.register.hp.com](http://www.register.hp.com)

본 설명서 및 설명서에 포함된 예시는 '있는 그대로' 제공되며 사전 통지 없이 변경될 수 있습니다. Hewlett-Packard Company는 본 설명서와 관련하여 상업성, 비침해성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(이에 제한되지 않음) 어떤 보증도 하지 않습니다.

Hewlett-Packard Company는 이 설명서 또는 여기에 포함된 예시의 제공, 성능 또는 사용과 관련하여 발생하는 우발적 또는 결과적 손해에 대해서도 책임을 지지 않습니다.

© 1988, 1990-1991, 2003, 2007 Hewlett-Packard Development Company, L.P. 저작권법에 따라 허용된 경우를 제외하고는 Hewlett-Packard Company의 사전 서면 승인 없이 본 설명서를 복제하거나, 수정하거나, 번역할 수 없습니다.

Hewlett-Packard Company  
16399 West Bernardo Drive  
MS 8-600  
San Diego, CA 92127-1899  
USA

---

## 인쇄 내역

제 1판

2007년 2월

# 목차

---

## 1부. 기본 연산

<b>1. 시작</b> .....	<b>1-1</b>
중요한 사전 준비 .....	1-1
계산기 켜기/끄기 .....	1-1
디스플레이 대비 조정 .....	1-1
키보드 및 디스플레이 요점 .....	1-2
Shift 키 .....	1-2
알파 키 .....	1-3
커서 키 .....	1-3
백스페이스 및 지우기 .....	1-4
메뉴 사용 .....	1-6
메뉴 종료 .....	1-8
RPN 및 ALG 모드 .....	1-9
실행 취소 키 .....	1-11
디스플레이 및 표시기 .....	1-12
숫자 입력 .....	1-15
숫자를 음수로 설정 .....	1-15
10의 지수 .....	1-15
입력 커서 이해 .....	1-17
숫자 범위 및 OVERFLOW .....	1-17
산술 계산 수행 .....	1-18
단일 인수 또는 단항 연산 .....	1-18
두 인수 또는 2항 연산 .....	1-19
디스플레이 형식 제어 .....	1-21
숫자의 마침표 및 쉼표(·) (,) .....	1-23

복소수 디스플레이 형식( $x+iy$ , $x+iy$ , $r\theta a$ ) .....	1-24
12자리의 전체 자릿수 표시 .....	1-25
분수 .....	1-26
분수 입력 .....	1-26
메시지 .....	1-27
계산기 메모리 .....	1-28
사용 가능한 메모리 확인 .....	1-28
모든 메모리 지우기 .....	1-29
<b>2. RPN: 자동 메모리 스택 .....</b>	<b>2-1</b>
스택 정의 .....	2-1
디스플레이 내의 X 레지스터와 Y 레지스터 .....	2-3
X 레지스터 지우기 .....	2-3
스택 검토 .....	2-3
스택에서 X 레지스터와 Y 레지스터 교환 .....	2-4
산술 - 스택의 산술 수행 방법 .....	2-5
ENTER 작동 방식 .....	2-6
스택을 지우는 방법 .....	2-7
LAST X 레지스터 .....	2-8
LAST X를 사용하여 오류 수정 .....	2-9
LAST X를 사용하여 숫자 다시 사용 .....	2-10
RPN 모드의 연속 계산 .....	2-12
괄호 없이 작업 .....	2-12
연습 .....	2-14
계산 순서 .....	2-14
추가 연습 .....	2-16
<b>3. 변수에 데이터 저장 .....</b>	<b>3-1</b>
숫자 저장 및 다시 호출 .....	3-2
변수 보기 .....	3-4

MEM 카탈로그 사용 .....	3-4
VAR 카탈로그 .....	3-4
저장된 변수를 사용한 산술 .....	3-6
저장 산술 .....	3-6
다시 호출 산술 .....	3-7
x를 임의 변수와 교환 .....	3-8
변수 "I" 및 "J" .....	3-9

#### **4. 실수 함수 .....4-1**

지수 및 로그 함수 .....	4-1
나눗셈의 몫과 나머지 .....	4-2
거듭제곱 함수 .....	4-2
삼각법 .....	4-3
$\pi$ 입력 .....	4-3
각도 모드 설정 .....	4-4
삼각 함수 .....	4-4
쌍곡선 함수 .....	4-6
백분율 함수 .....	4-6
물리상수 .....	4-8
변환 함수 .....	4-10
직교좌표/극좌표 변환 .....	4-10
시간 변환 .....	4-13
각도 변환 .....	4-13
단위 변환 .....	4-14
확률 함수 .....	4-15
계승 .....	4-15
감마 .....	4-15
확률 .....	4-15
숫자의 특정 부분 .....	4-17

<b>5. 분수</b> .....	<b>5-1</b>
분수 입력 .....	5-1
디스플레이의 분수 .....	5-2
표시 규칙 .....	5-2
정확도 표시기 .....	5-3
분수 표시 변경 .....	5-4
최대 분모 설정 .....	5-4
분수 형식 선택 .....	5-6
분수 표시 예 .....	5-8
분수 반올림 .....	5-8
방정식의 분수 .....	5-9
프로그램의 분수 .....	5-10
<b>6. 방정식 입력 및 계산</b> .....	<b>6-1</b>
방정식을 사용할 수 있는 방법 .....	6-1
방정식 연산 요약 .....	6-3
방정식 목록에 방정식 입력 .....	6-4
방정식의 변수 .....	6-4
방정식의 숫자 .....	6-5
방정식의 함수 .....	6-5
방정식의 괄호 .....	6-6
방정식 표시 및 선택 .....	6-6
방정식 편집 및 지우기 .....	6-8
방정식 유형 .....	6-9
방정식 계산 .....	6-10
계산을 위해 ENTER 사용 .....	6-11
계산을 위해 XEQ 사용 .....	6-12
방정식 프롬프트에 응답 .....	6-13
방정식 구문 .....	6-14

연산자 우선순위 .....	6-14
방정식 함수 .....	6-16
구문 오류 .....	6-19
방정식 확인 .....	6-19
<b>7. 방정식 풀기.....</b>	<b>7-1</b>
방정식 풀기.....	7-1
기본 제공 방정식 풀기 .....	7-6
SOLVE 이해 및 제어 .....	7-7
결과 확인 .....	7-7
SOLVE 계산 일시 중단 .....	7-8
SOLVE에 대한 초기 추정값 선택 .....	7-8
추가 정보 .....	7-12
<b>8. 방정식 적분.....</b>	<b>8-1</b>
방정식 적분 ( $\int FN$ ).....	8-2
적분의 정확도 .....	8-6
정확도 지정 .....	8-6
정확도 해석 .....	8-6
추가 정보 .....	8-8
<b>9. 복소수 연산.....</b>	<b>9-1</b>
복소수 스택 .....	9-2
복소수 연산 .....	9-2
극좌표 표기법으로 복소수 사용.....	9-6
방정식의 복소수 .....	9-8
프로그램의 복소수 .....	9-9
<b>10. 산술 벡터 .....</b>	<b>10-1</b>
벡터 연산 .....	10-1
벡터의 절대값.....	10-3

내적 .....	10-4
벡터 사이의 각도 .....	10-5
방정식의 벡터 .....	10-6
프로그램의 벡터 .....	10-6
변수 또는 레지스터에서 벡터 만들기 .....	10-7
<b>11.진법 변환과 산술 및 논리 .....</b>	<b>11-1</b>
2, 8 및 16진수의 산술 .....	11-4
숫자의 표현 .....	11-6
음수 .....	11-6
숫자 범위 .....	11-7
긴 2진수의 구간 .....	11-8
프로그램 및 방정식에 진법 사용 .....	11-8
<b>12.통계 연산 .....</b>	<b>12-1</b>
통계 데이터 입력 .....	12-1
1개 변수 데이터 입력 .....	12-2
2개 변수 데이터 입력 .....	12-2
데이터 입력의 오류 수정 .....	12-2
통계 계산 .....	12-4
평균 .....	12-4
표본 표준 편차 .....	12-6
모집단 표준 편차 .....	12-7
선형 회귀 .....	12-7
데이터 정밀도의 한계 .....	12-11
합계 값 및 통계 레지스터 .....	12-12
합계 통계 .....	12-12
통계 레지스터 액세스 .....	12-13



## 2부. 프로그래밍

<b>13.단순 프로그래밍</b> .....	<b>13-1</b>
프로그램 디자인 .....	13-3
모드 선택 .....	13-3
프로그램 경계(LBL 및 RTN) .....	13-4
프로그램에 RPN, ALG 및 방정식 사용 .....	13-4
데이터 입력 및 출력 .....	13-5
프로그램 입력 .....	13-6
지우기 기능 및 백스페이스 키 .....	13-7
프로그램의 기능 이름.....	13-8
프로그램 실행 .....	13-10
프로그램 실행(XEQ) .....	13-10
프로그램 테스트 .....	13-11
데이터 입력 및 표시 .....	13-12
데이터 입력에 INPUT 사용 .....	13-13
데이터 표시에 VIEW 사용 .....	13-15
방정식을 사용하여 메시지 표시.....	13-16
중지하지 않고 정보 표시.....	13-18
프로그램 중지 또는 일시 중단 .....	13-19
중지 또는 일시 중지(STOP, PSE) 프로그래밍 .....	13-19
실행 중인 프로그램 일시 중단 .....	13-19
오류 중지 .....	13-19
프로그램 편집 .....	13-20
프로그램 메모리 .....	13-21
프로그램 메모리 보기 .....	13-21
메모리 사용 .....	13-22
프로그램 카탈로그(MEM).....	13-22

한 개 또는 여러 개의 프로그램 지우기 .....	13-23
체크섬 .....	13-23
프로그램밍할 수 없는 기능 .....	13-24
BASE를 사용하여 프로그래밍 .....	13-25
프로그램에서 진법 모드 선택 .....	13-25
프로그램 줄에 입력된 숫자 .....	13-25
다항식 및 호너(Horner) 방법 .....	13-26

## 14. 프로그래밍 기술..... 14-1

프로그램의 루틴 .....	14-1
서브루틴 호출(XEQ, RTN) .....	14-1
중첩된 서브루틴 .....	14-2
분기(GTO) .....	14-5
프로그래밍된 GTO 명령 .....	14-5
키보드에서 GTO 사용 .....	14-6
조건부 명령 .....	14-6
비교 테스트( $x?y$ , $x?0$ ) .....	14-7
플래그 .....	14-9
루프 .....	14-16
조건부 루프(GTO) .....	14-16
카운터를 사용하여 루프(DSE, ISG) .....	14-17
변수 및 레이블에 간접 주소 지정 .....	14-19
변수 "I" 및 "J" .....	14-20
간접 주소 (I) 및 (J) .....	14-20
(I)/(J)를 사용한 프로그램 제어 .....	14-21
(I)/(J)를 사용한 방정식 .....	14-22
명명되지 않은 간접 변수 .....	14-22

<b>15. 프로그램 값 구하기 및 적분 .....</b>	<b>15-1</b>
프로그램 값 구하기 .....	15-1
프로그램에 SOLVE 사용 .....	15-6
프로그램 적분 .....	15-8
프로그램에 적분 사용 .....	15-10
값 구하기와 적분의 한계 .....	15-11
<b>16. 통계 프로그램 .....</b>	<b>16-1</b>
곡선 맞춤 .....	16-1
정규 분포 및 역 정규 분포 .....	16-12
그룹화된 표준 편차 .....	16-18
<b>17. 기타 프로그램 및 방정식 .....</b>	<b>17-1</b>
화폐의 시간적 가치 .....	17-1
소수 생성기 .....	17-7
벡터의 외적 .....	17-11

### **3부. 부록 및 참조**

<b>A. 지원, 배터리 및 서비스 .....</b>	<b>A-1</b>
계산기 지원 .....	A-1
일반적인 질문에 대한 답변 .....	A-1
환경적인 제한 사항 .....	A-2
배터리 교체 .....	A-3
계산기 작동 테스트 .....	A-4
자가 테스트 .....	A-5
보증 .....	A-7
고객 지원 .....	A-9
Regulatory information .....	A-13
Federal Communications Commission Notice .....	A-13

<b>B. 사용자 메모리 및 스택</b> .....	<b>B-1</b>
계산기 메모리 관리 .....	B-1
계산기 재설정 .....	B-3
메모리 지우기 .....	B-3
스택 올리기의 상태 .....	B-5
해제 연산.....	B-5
중립 연산.....	B-5
LAST X 레지스터의 상태 .....	B-6
스택 레지스터 내용 액세스 .....	B-7
 <b>C. ALG: 요약</b> .....	 <b>C-1</b>
ALG 정보.....	C-1
ALG에서 <b>두 인수</b> 산술 실행 .....	C-2
단순 산술.....	C-2
거듭제곱 함수 .....	C-3
백분율 계산 .....	C-3
순열 및 조합 .....	C-4
나눗셈의 몫과 나머지 .....	C-4
괄호 계산 .....	C-5
지수 및 로그 함수.....	C-5
삼각 함수 .....	C-6
쌍곡선 함수 .....	C-6
숫자의 특정 부분 .....	C-7
스택 검토 .....	C-7
방정식 적분 .....	C-8
복소수 연산 .....	C-8
2, 8 및 16진수의 산술 .....	C-10
2개 변수 통계 데이터 입력 .....	C-11

<b>D. 방정식 풀기 추가 정보</b> .....	<b>D-1</b>
SOLVE에서 근을 구하는 방법 .....	D-1
결과 해석 .....	D-3
SOLVE에서 근을 구할 수 없는 경우.....	D-8
반올림 오차.....	D-13
<b>E. 적분 추가 정보</b> .....	<b>E-1</b>
적분 계산 방법 .....	E-1
잘못된 결과가 나올 수 있는 경우.....	E-2
계산 시간이 길어지는 경우 .....	E-7
<b>F. 메시지</b> .....	<b>F-1</b>
<b>G. 연산 색인</b> .....	<b>G-1</b>
<b>H. 색인</b> .....	<b>H-1</b>



# 1부

## 기본 연산

---





## 시작

---



여백에 있는 이 기호는 RPN 모드로 표시되며 ALG 모드에서는 다르게 수행되어야 하는 키 입력이나 예를 나타냅니다.

부록 C에서는 ALG 모드에서 계산기를 사용하는 방법에 대해 설명합니다.

---

## 중요한 사전 준비

### 계산기 켜기/끄기

계산기를 켜려면 **[C]**를 누릅니다. **[C]** 키의 아래쪽에 ON이 인쇄됩니다.

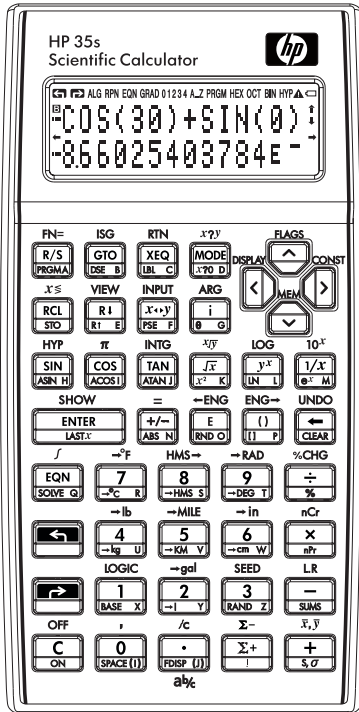
계산기를 끄려면 **[ON]** **[C]**를 누릅니다. 즉, **[ON]** shift 키를 눌렀다가 놓은 후 위에 노란색으로 OFF가 인쇄되어 있는 **[C]**를 누릅니다. 계산기에는 연속 메모리가 있어서 계산기를 꺼도 저장된 정보는 지워지지 않습니다.

에너지를 절약하기 위해 10분 이상 사용하지 않으면 계산기가 자동으로 꺼집니다. 디스플레이에 배터리 부족 표시(**[BATT]**)가 나타나면 최대한 빨리 배터리를 교체합니다. 자세한 내용은 부록 A를 참조하십시오.

### 디스플레이 대비 조정





디스플레이 대비는 조명, 시각 및 디스플레이 설정에 따라 달라집니다. 대비를 늘리거나 줄이려면 **[C]** 키를 누른 상태로 **[+]** 또는 **[-]**를 누릅니다.

# 키보드 및 디스플레이 요점



## Shift 키

각 키에는 세 가지 기능, 즉 정면에 인쇄된 기능, 왼쪽 shift 기능(노란색) 및 오른쪽 shift 기능(파란색)이 있습니다. *shift* 기능 이름은 각 키의 위에 노란색으로 인쇄되고 아래쪽에 파란색으로 인쇄됩니다. 원하는 기능의 키를 누르기 전에 해당 shift 키(☞ 또는 ☞)를 누릅니다. 예를 들어, 계산기를 끄려면 ☞ shift 키를 눌렀다가 놓은 후 ☞를 누릅니다.

 또는 를 누르면 디스플레이 맨 위의 해당  또는  표시기 기호가 켜집니다. 이 표시기는 다음 키를 누를 때까지 켜져 있습니다. shift 키를 취소하고 표시기를 끄려면 동일한 shift 키를 다시 누릅니다.





## 알파 키



대부분의 키는 위와 같이 오른쪽 아래에 글자를 표시합니다. 글자(예: 변수 또는 프로그램 레이블)를 입력해야 할 때마다 **A..Z** 표시기가 디스플레이에 표시되어 알파 키가 "활성" 상태임을 나타냅니다.

변수에 대해서는 3장에서 설명하고, 레이블에 대해서는 13장에서 설명합니다.











## 커서 키

4개의 커서 방향 키는 각각 화살표로 표시됩니다. 이 텍스트에서는 , ,  및  그래픽을 사용하여 해당 키를 나타냅니다.


# 백스페이스 및 지우기

우선 입력을 지우고, 숫자를 수정하고, 처음부터 시작하기 위해 전체 디스플레이를 지우는 방법을 알아야 합니다.

## 지우기 키

키	설명
	<p><i>백스페이스 키입니다.</i></p> <p>수식을 입력 중인 경우 는 입력 커서( <code>_</code> ) 왼쪽의 문자를 지웁니다. 그렇지 않으면 완성된 수식이나 줄 2의 계산 결과에 대해 가 해당 결과를 0으로 대체합니다. 또한 는 오류 메시지를 지우고 메뉴를 종료합니다. 는 아래에 설명된 것처럼 계산기가 프로그램 입력 모드와 방정식 입력 모드에 있을 때 유사하게 동작합니다.</p> <ul style="list-style-type: none"> <li>■ 방정식 입력 모드: 방정식을 입력 또는 편집 중인 경우 는 삽입 커서 왼쪽의 문자를 지웁니다. 그렇지 않고 방정식이 입력된 경우(삽입 커서가 없음) 는 전체 방정식을 삭제합니다.</li> <li>■ 프로그램 입력 모드: 프로그램 줄을 입력 또는 편집 중인 경우 는 삽입 커서 왼쪽의 문자를 지웁니다. 그렇지 않고 프로그램 줄이 입력된 경우 는 전체 줄을 삭제합니다.</li> </ul>
	<p><i>지우기 또는 취소입니다.</i></p> <p>표시된 숫자를 0으로 지우거나 현재 상황(예: 메뉴, 메시지, 프롬프트, 카탈로그, 방정식 입력 모드 또는 프로그램 입력 모드)을 취소합니다.</p>

## 지우기 키(계속)


키	설명
 CLEAR	<p><i>CLEAR</i> 메뉴(<math>\times</math> VARS ALL <math>\Sigma</math> STK CLVAR<math>\times</math>)에는 <math>\times</math>(X 레지스터의 숫자), 모든 직접 변수, 모든 메모리, 모든 통계 데이터, 모든 스택 및 간접 변수를 지우는 옵션이 포함되어 있습니다.</p> <p><b>3</b> (3ALL)을 누르면 새 메뉴 CLR ALL? Y N이 표시되어 메모리에 있는 모든 내용을 지우기 전에 결정을 확인할 수 있습니다.</p> <p>프로그램 입력 중에 3ALL은 3PGM으로 대체됩니다.</p> <p><b>3</b> (3PGM)을 누르면 새 메뉴 CLR PGMS? Y N이 표시되어 모든 프로그램을 지우기 전에 결정을 확인할 수 있습니다.</p> <p>방정식 입력 중에 3ALL은 3EQN으로 대체됩니다.</p> <p><b>3</b> (3EQN)을 누르면 CLR EQN? Y N 메뉴가 표시되어 모든 방정식을 지우기 전에 결정을 확인할 수 있습니다.</p> <p><b>6</b> (CLVAR<math>\times</math>)을 선택하면 3개의 자리 표시자가 있는 명령줄에 명령이 붙여넣어집니다. 자리 표시자 공백에 3자리 숫자를 입력해야 합니다. 그러면 주소가 입력한 주소보다 큰 간접 변수는 모두 지워집니다. 예를 들어, CLVAR056은 주소가 56보다 큰 간접 변수를 모두 지웁니다.</p>

# 메뉴 사용





HP 35s에는 키보드에 표시되는 것보다 훨씬 더 많은 기능이 있습니다. 이것은 16개의 키가 메뉴 키이기 때문입니다. 모두 16개의 메뉴가 있으며, 훨씬 더 많은 기능이나 기능 옵션을 제공합니다.

## HP 35s 메뉴

메뉴 이름	메뉴 설명	장
	<b>숫자 기능</b>	
L.R.	$\hat{x} \hat{y} r m b$ 선형 회귀: 곡선 맞춤 및 선형 추산입니다.	12
$\bar{x}, \bar{y}$	$\bar{x} \bar{y} \bar{x} \bar{y}$ 통계 $x$ 값과 $y$ 값의 산술 평균, 통계 $x$ 값의 가중 평균입니다.	12
$s, \sigma$	$s \times s \sigma \times \sigma$ 표본 표준 편차, 모집단 표준 편차입니다.	12
CONST	41개의 상수 값에 액세스하는 메뉴입니다. 4-8페이지의 "물리상수"를 참조하십시오.	4
SUMS	$n \Sigma x \Sigma y \Sigma x^2 \Sigma y^2 \Sigma xy$ 통계 데이터 합계입니다.	12
BASE	DEC HEX OCT BIN $d h o b$ 진법 변환(10진, 16진, 8진 및 2진)입니다.	12
INTG	SGN INT÷ Rmdr INTG FP IP 부호 값, 정수 나누기, 나눗셈의 나머지, 최대 정수, 소수부, 정수부입니다.	4,C
LOGIC	AND XOR OR NOT NAND NOR 논리 연산자입니다.	11

<b>프로그래밍 명령</b>		
FLAGS	SF CF FS? 플래그를 설정하고 지우고 테스트하는 기능입니다.	14
x?y	$\neq \leq < > \geq =$ X 및 Y 레지스터의 비교 테스트입니다.	14
x?0	$\neq \leq < > \geq =$ X 레지스터와 0의 비교 테스트입니다.	14
<b>기타 기능</b>		
MEM	VAR PGM 메모리 상태(사용 가능한 메모리 바이트), 변수 카탈로그, 프로그램(프로그램 레이블) 카탈로그입니다.	1, 3, 12
MODE	DEG RAD GRAD ALG RPN 각도 모드와 연산 모드입니다.	4, 1
디스플레이	FIX SCI ENG ALL. , 1,000 1000 x <i>iv</i> x+y <i>iv</i> r i i i $\neq$ 고정, 과학, 공학, 전체 부동 소수점 수 디스플레이 형식, 기수 기호 옵션(. 또는 ,), 복소수 디스플레이 형식(RPN 모드에서는 xiy 및 rqa만 사용 가능)입니다.	1
R↓ R↑	X Y Z T ALG 모드에서 스택을 검토하는 기능(X, Y, Z, T 레지스터)입니다.	C
CLEAR	메모리의 다른 부분을 지우는 기능입니다. 1.5페이지에 있는 테이블의  CLEAR를 참조하십시오.	1, 3, 6, 12

**메뉴 기능을 사용하는 방법은 다음과 같습니다.**

1. 메뉴 키를 눌러 메뉴 항목 세트를 표시합니다.
2.    를 눌러 선택하려는 항목으로 밑줄을 이동합니다.
3. 항목에 밑줄이 있는 동안 **ENTER**를 누릅니다.

번호가 매겨진 메뉴 항목의 경우 항목에 밑줄이 있는 동안 **ENTER**를 누르거나 항목 번호를 입력하면 됩니다.

CONST 및 SUMS과 같은 일부 메뉴에는 여러 페이지가 있습니다. 이러한 메뉴를 시작하면 **▲** 또는 **▼** 표시기가 켜집니다. 이 메뉴에서는 **▶** 및 **◀** 커서 키를 사용하여 현재 메뉴 페이지의 항목으로 이동하고, **▼** 및 **▲** 키를 사용하여 메뉴의 다음 및 이전 페이지에 액세스합니다.

**예:**

이 예에서는 DISPLAY 메뉴를 사용하여 숫자 표시를 소수 네 자릿수로 고정된 다음 6÷7을 계산합니다. 그리고 DISPLAY 메뉴 사용을 닫아 전체 부동 소수점 숫자 표시로 돌아갑니다.

키:	디스플레이:	설명:
	0	초기 표시입니다.
	0	
<b>◀</b> <b>DISPLAY</b>	<b>1FIX</b> 2SCI	DISPLAY 메뉴를 시작합니다.
	3ENG 4ALL	
<b>1</b> 또는 <b>ENTER</b>	FIX _	Fix 명령이 줄 2에 붙여넣어집니다.
<b>4</b>	0.0000	소수 네 자릿수로 고정됩니다.
	0.0000	
<b>6</b> <b>ENTER</b> <b>7</b> <b>↵</b>	0.0000	나눗셈을 수행합니다.
	0.8571	
<b>◀</b> <b>DISPLAY</b> <b>4</b>	0	전체 자릿수로 돌아갑니다.
	8.57142857143E-	

메뉴는 수많은 기능에 대한 가이드를 제공하며, 각 기능을 실행하는 데 도움을 줍니다. 따라서 계산기에 내장된 모든 기능의 이름을 기억할 필요는 없으며 키보드에 인쇄된 기능을 검색할 필요도 없습니다.

**메뉴 종료**

위의 예와 같이 메뉴 기능을 실행할 때마다 메뉴가 자동으로 사라집니다. 기능을 실행하지 *않고* 메뉴를 그대로 두려는 경우 다음 세 가지 옵션을 사용할 수 있습니다.



- 를 누르면 한 번에 한 수준씩, 2 수준 CLEAR 또는 MEM 메뉴에서 뒤로 이동합니다. 1-5페이지에 있는 테이블의 (CLEAR)를 참조하십시오.
- 또는 를 누르면 메뉴가 취소됩니다.

키:	디스플레이:
	123.5678_
	<u>1FIX</u> 2SCI    ↓ 3ENG    4ALL
or	123.5678_

- 다른 메뉴 키를 누르면 기존 메뉴에서 새로운 메뉴로 전환됩니다.

키:	디스플레이:
	123.5678_
	<u>1FIX</u> 2SCI    ↓ 3ENG    4ALL
	<u>1X</u> 2VARS    ↓ 3ALL    4Σ
	123.5678

## RPN 및 ALG 모드

RPN(역 폴란드 표기법) 모드나 ALG(대수) 모드로 산술 연산을 수행하도록 계산기를 설정할 수 있습니다.

RPN(역 폴란드 표기법) 모드에서는 계산의 중간 결과가 자동으로 저장되므로 괄호를 사용할 필요가 없습니다.

ALG(대수) 모드에서는 표준 연산 순서를 사용하여 산술 연산을 수행합니다.

### RPN 모드를 선택하려면:

(5RPN)을 눌러 계산기를 RPN 모드로 설정합니다. 계산기가 RPN 모드로 되어 있으면 **RPN** 표시기가 켜집니다.

## ALG 모드를 선택하려면:

**[MODE]** **[4]** (4ALG)를 눌러 계산기를 ALG 모드로 설정합니다. 계산기가 ALG 모드로 되어 있으면 **ALG** 표시기가 켜집니다.

## 예:

1 + 2 = 3을 계산하려고 합니다.

RPN 모드에서는 첫 번째 숫자를 입력하고 **[ENTER]** 키를 누른 다음 두 번째 숫자를 입력하고 마지막으로 산술 연산자 키 **[+]**를 누릅니다.

ALG 모드에서는 첫 번째 숫자를 입력하고 **[+]**를 누른 다음 두 번째 숫자를 입력하고 마지막으로 **[ENTER]** 키를 누릅니다.

RPN 모드	ALG 모드
1 <b>[ENTER]</b> 2 <b>[+]</b>	1 <b>[+]</b> 2 <b>[ENTER]</b>

ALG 모드에서는 결과와 계산이 표시됩니다. RPN 모드에서는 결과만 표시되고 계산은 표시되지 않습니다.

---

### 참고






계산을 수행할 때 ALG(대수) 모드나 RPN(역 폴란드 표기법) 모드 중에서 선택할 수 있습니다. 설명서 전체에서 여백의 “✓”는 RPN 모드의 예나 키 입력이 ALG 모드에서는 다르게 수행되어야 함을 나타냅니다. 부록 C에서는 ALG 모드에서 계산기를 사용하는 방법에 대해 설명합니다.


---




# 실행 취소 키

## 실행 취소 키

실행 취소 키의 동작은 계산기 컨텍스트에 따라 달라지지만 주로 임의 연산을 실행 취소하는 것이 아니라 입력 삭제를 복구하는 데 사용됩니다. 숫자 함수가 실행된 후 디스플레이의 줄 2에서 입력을 취소하는 방법에 대한 자세한 내용은 *마지막 X 레지스터*를 참조하십시오. 다음을 복구하려면  또는 를 사용한 직후에  UNDO를 누릅니다.

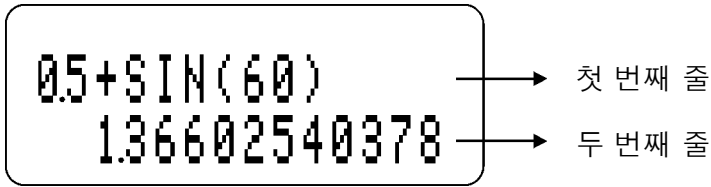
- 삭제한 입력
- 방정식 모드에 있는 동안 삭제한 방정식
- 프로그램 모드에 있는 동안 삭제한 프로그램 줄

또한 명령 취소를 사용하여 CLEAR 메뉴를 통해 방금 지운 레지스터 값을 복구할 수 있습니다. 명령 취소 작업은 삭제 작업 바로 뒤에 수행되어야 하며, 중간에 다른 작업을 수행하면 명령 취소를 통해 삭제된 객체를 복구할 수 없습니다. 삭제 후 전체 입력을 복구할 뿐 아니라 입력을 편집하는 동안 명령 취소를 사용할 수도 있습니다. 다음을 복구하려면 편집 중에  UNDO를 누릅니다.

- 를 사용하여 방금 삭제한 수식의 숫자
- 편집 중이었지만 를 사용하여 지운 수식
- 방정식 모드나 프로그램 모드에 있는 동안 를 사용하여 방금 삭제한 방정식 또는 프로그램의 문자

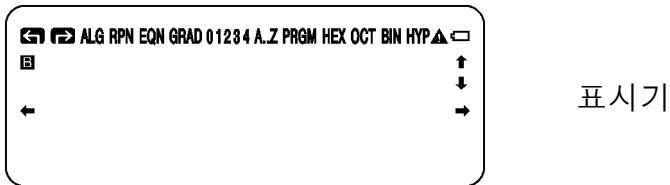
명령 취소 작업은 사용 가능한 메모리 크기에 의해 제한됩니다.

## 디스플레이 및 표시기



디스플레이는 두 줄과 표시기로 이루어져 있습니다.

14자 이상의 입력은 왼쪽으로 스크롤됩니다. 입력 중에 ALG 모드에서는 입력이 첫 번째 줄에 표시되고 RPN 모드에서는 두 번째 줄에 표시됩니다. 모든 계산은 E 부호(지수)와 최대 3자리수의 지수 값을 포함하여 최대 14자리수로 표시됩니다.












위의 그림에 표시된 디스플레이의 기호를 표시기라고 합니다. 각 표시기는 디스플레이에 표시될 때 특별한 의미를 나타냅니다.

## HP 35s 표시기

표시기	의미	장
<b>B</b>	“ <b>B</b> (사용 중)” 표시기는 연산, 방정식 또는 프로그램이 실행되는 동안 나타납니다.	
<b>▲</b> <b>▼</b>	분수 디스플레이 모드( <b>FR</b> <b>FDISP</b> 누름)에 있을 때는 “ <b>▲▼</b> ” 표시기의 절반인 “ <b>▲</b> ” 또는 “ <b>▼</b> ” 중 하나만 켜져 표시된 분자가 실제 값보다 조금 더 작거나 크을 나타냅니다. “ <b>▲▼</b> ”의 어떤 부분도 켜지지 않으면 분수의 값이 <i>정확하게</i> 표시되고 있습니다.	5
<b>⇐</b>	왼쪽 shift가 활성화 상태입니다.	1
<b>⇒</b>	오른쪽 shift가 활성화 상태입니다.	1
<b>RPN</b>	역 폴란드 표기법 모드가 활성화 상태입니다.	1, 2
<b>ALG</b>	대수 모드가 활성화 상태입니다.	1, C
<b>PRGM</b>	프로그램 입력이 활성화 상태입니다.	13
<b>EQN</b>	방정식 입력 모드가 활성화 상태이거나, 계산기가 수식을 계산 중이거나 방정식을 실행 중입니다.	6
<b>0 1 2 3 4</b>	설정된 플래그를 나타냅니다(플래그 5-11에는 표시기가 없음).	14
<b>RAD</b> 또는 <b>GRAD</b>	라디안 또는 그레드 각도 모드가 설정되었습니다. DEG 모드(기본값)에는 표시기가 없습니다.	4
<b>HEX OCT BIN</b>	활성 숫자 밑을 나타냅니다. DEG(밑 10, 기본값)에는 표시기가 없습니다.	11
<b>HYP</b>	쌍곡선 함수가 활성화 상태입니다.	4, C

## HP 35s 표시기(계속)

표시기	의미	장
←, →	입력 디스플레이에서 줄 1 또는 줄 2의 왼쪽이나 오른쪽에 추가 문자가 있습니다. 두 표시기는 동시에 나타날 수 있으며 입력 디스플레이의 왼쪽과 오른쪽에 문자가 있음을 나타냅니다. 누락된 문자가 있는 줄 1의 입력에는 줄 임표(...)가 표시되어 누락된 문자를 나타냅니다. RPN 모드에서는  및  키를 사용하여 입력을 스크롤하고 선행 및 후행 문자를 표시합니다. ALG 모드에서는   및   를 사용하여 나머지 문자를 표시합니다.	1, 6
↑, ↓	 및  키는 방정식 목록, 변수 카탈로그, 프로그램 줄, 메뉴 페이지 또는 프로그램 카탈로그에 있는 프로그램의 단계별 이동을 위해 활성화됩니다.	1, 6, 13
A..Z	알파벳 키가 활성 상태입니다.	3
▲	주의! 특수 상태나 오류를 나타냅니다.	1
	배터리 전원이 부족합니다.	A



키:	디스플레이:	설명:
<b>0</b> <b>.</b> <b>0</b> <b>0</b>	0.000062_	입력되는 숫자를 표시합니다.
<b>0</b> <b>0</b> <b>6</b> <b>2</b>	0.0001	
<b>ENTER</b>		디스플레이 형식에 맞게 숫자를 반올림합니다.
<b>0</b> <b>.</b> <b>0</b> <b>0</b>	4.2000E-5	다른 방법으로는 유효 숫자가 표시되지 않으므로 자동으로 유효숫자 표기법을 사용합니다.
<b>0</b> <b>0</b> <b>4</b> <b>2</b>		
<b>ENTER</b>		

## 10의 거듭제곱 입력

**E** 키는 10의 거듭제곱을 신속하게 입력하는 데 사용됩니다. 예를 들어, 백만을 1000000으로 입력하는 대신 **1** **E** **6**을 입력할 수 있습니다. 다음 예에서는 이 프로세스와 계산기에서 결과를 표시하는 방법을 보여 줍니다.

### 예:

플랑크 상수인  $6.6261 \times 10^{-34}$ 을 입력한다고 가정합니다.

키:	디스플레이:	설명
<b>6</b> <b>.</b> <b>6</b> <b>2</b> <b>6</b>	0	가수를 입력합니다.
<b>1</b>	6.6261_	
<b>E</b>	0	$\times 10^x$ 에 해당합니다.
	6.621E_	
<b>3</b> <b>4</b> <b>+/-</b> <b>ENTER</b>	6.621E <sup>-34</sup>	지수를 입력합니다.
	6.621E-34	

위의 백만 예와 같이 승수가 없는 10의 거듭제곱의 경우 **1** **E** 키를 누른 다음 원하는 10의 지수를 입력합니다.



## 기타 지수 기능

10의 지수(밑 10 역대수)를 계산하려면  $\boxed{\frac{1}{x}}$   $\boxed{10^x}$ 를 사용합니다. *임의* 숫자의 거듭제곱(누승법) 결과를 계산하려면  $\boxed{y^x}$ 를 사용합니다(4장 참조).

## 입력 커서 이해

숫자를 입력할 때 커서( )가 나타나고 디스플레이에서 깜박입니다. 커서는 다음 숫자의 이동 위치를 보여 주므로 숫자가 완성되지 않았음을 나타냅니다.

키:	디스플레이:	설명:
$\boxed{1} \boxed{2} \boxed{3}$	123_	입력이 종료되지 <i>않았습니다</i> . 숫자가 완성되지 <i>않았습니다</i> .

*함수를 실행하여 결과를* 계산하는 경우 숫자가 완성되고 입력이 종료되었으므로 커서가 사라집니다.

✓  $\boxed{\sqrt{x}}$  11.0905      입력이 종료되었습니다.

$\boxed{\text{ENTER}}$ 를 누르면 입력이 종료됩니다. 두 숫자를 구분하려면 첫 번째 숫자를 입력하고  $\boxed{\text{ENTER}}$ 를 눌러 입력을 종료한 다음 두 번째 숫자를 입력합니다.

✓  $\boxed{1} \boxed{2} \boxed{3} \boxed{\text{ENTER}}$  123.0000      완성된 숫자입니다.

✓  $\boxed{4} \boxed{+}$  127.0000      다른 완성된 숫자입니다.

입력이 종료되지 *않은* 경우(커서가 있는 경우)  $\boxed{\leftarrow}$ 는 백스페이스하여 마지막 숫자를 지웁니다. 입력이 종료된 경우(커서 없음)  $\boxed{\leftarrow}$ 는  $\boxed{C}$ 처럼 동작하며 전체 숫자를 지웁니다. 시도해 보십시오.

## 숫자 범위 및 OVERFLOW

계산기에서 사용 가능한 가장 작은 숫자는  $-9.9999999999 \times 10^{499}$ 이고, 가장 큰 숫자는  $9.9999999999 \times 10^{499}$ 입니다.

- 계산 결과가 가능한 최대 숫자를 초과하면  $-9.9999999999 \times 10^{499}$  또는  $9.9999999999 \times 10^{499}$ 이 반환되며 **OVERFLOW**라는 경고 메시지가 나타납니다.

---

## 산술 계산 수행

HP 35s는 RPN 모드나 대수 모드(ALG)로 작동할 수 있습니다. 두 모드는 수식의 입력 방법에 영향을 줍니다. 다음 절에서는 단일 인수(또는 단항) 및 두 인수(또는 2항) 연산의 입력 차이점에 대해 설명합니다.

### 단일 인수 또는 단항 연산

$1/x$ ,  $x^2$ ,  $\ln$  및  $\sin$ 과 같은 HP 35s의 일부 숫자 연산에서는 단일 숫자의 입력이 필요합니다. 이러한 단일 인수 연산은 계산기가 RPN 모드에 있는지, 또는 ALG 모드에 있는지에 따라 다르게 입력됩니다. RPN 모드에서는 숫자를 먼저 입력한 다음 연산이 적용됩니다. 숫자를 입력한 후 **ENTER** 키를 누르면 숫자가 줄 1에 나타나고 결과가 줄 2에 표시됩니다. 그렇지 않으면 결과만 줄 2에 표시되고 줄 1은 변경되지 않습니다. ALG 모드에서는 연산자를 먼저 누르며 디스플레이에 함수와 괄호 세트가 표시됩니다. 괄호 사이에 숫자를 입력하고 **ENTER** 키를 누릅니다.

수식이 줄 1에 표시되고 결과가 줄 2에 표시됩니다. 다음 예에서는 차이점을 보여줍니다.

## 예:

먼저 RPN 모드에서  $3.4^2$ 를 계산한 다음 ALG 모드에서 계산합니다.

키:	디스플레이:	설명:
<b>MODE</b> <b>5</b> (5RPN)		RPN 모드를 시작합니다 (필요한 경우).
<b>3</b> <b>.</b> <b>4</b>	0 3.4	숫자를 입력합니다.
<b>→</b> <b>x<sup>2</sup></b>	0 11.56	제곱 연산자를 누릅니다.
<b>MODE</b> <b>4</b> (4ALG)		ALG 모드로 전환합니다.
<b>→</b> <b>x<sup>2</sup></b>	SQ()	제곱 연산을 입력합니다.
<b>3</b> <b>.</b> <b>4</b>	SQ(3.4)	괄호 사이에 숫자를 삽입합니다.
<b>ENTER</b>	SQ(3.4) 11.56	Enter 키를 눌러 결과를 표시합니다.

이 예에서 제곱 연산자는 키에 **x<sup>2</sup>**으로 나타나지만 SQ()로 표시됩니다. ALG 모드에서 키보드 및 RPN 모드와는 다르게 표시되는 몇 개의 단일 인수 연산자가 있습니다. 이러한 연산자는 아래 표에 나열되어 있습니다.

키	RPN, RPN 프로그램에서	ALG, 방정식, ALG 프로그램에서
<b>x<sup>2</sup></b>	$x^2$	SQ()
<b>√x</b>	$\sqrt{x}$	SQRT()
<b>e<sup>x</sup></b>	$e^x$	EXP()
<b>10<sup>x</sup></b>	$10^x$	ALOG()
<b>1/x</b>	$1/x$	INV()

## 두 인수 또는 2항 연산

**+**, **÷**, **y<sup>x</sup>** 및 **nCr**과 같은 두 인수 연산도 모드에 따라 다르게 입력됩니다. 차이점은 단일 인수 연산자의 경우와 유사합니다. RPN 모드에서는 첫 번째 숫자를 입력한 후 두 번째 숫자가 x 레지스터에 넣어지고 두 인수 연산이 호출됩니다. ALG 모드에서는 두 가지 경우, 즉 일반적인 사이 표기법을 사용하는 경우와 보다 함수 지향적인 방법을 사용하는 경우가 있습니다. 다음 예에서는 차이점을 보여줍니다.

## 예

먼저 RPN 모드에서  $2+3$  및  ${}_6C_4$ 를 계산한 다음 ALG 모드에서 계산합니다.

키:	디스플레이:	설명:
<b>MODE</b> <b>5</b> (5RPN)	2	RPN 모드로 전환합니다(필요한 경우).
<b>2</b> <b>ENTER</b> <b>3</b>	3_	2를 입력한 다음 x 레지스터에 3을 넣습니다. 3 뒤에 커서가 깜박입니다.
<b>+</b>	0	Enter 키는 누르지 마십시오. 더하기 키를 눌러 결과를 표시합니다.
<b>6</b> <b>ENTER</b> <b>4</b>	5 6	6을 입력한 다음 x 레지스터에 4를 넣습니다.
<b>↵</b> nCr	4_ 5_	조합 키를 눌러 결과를 표시합니다.
<b>MODE</b> <b>4</b> (4ALG)	15	ALG 모드로 전환합니다.
<b>2</b> <b>+</b> <b>3</b> <b>ENTER</b>	2+3	수식과 결과가 모두 표시됩니다.
<b>↵</b> nCr	5	조합 함수를 입력합니다.
<b>6</b> <b>&gt;</b> <b>4</b>	nCr(, ) nCr(6,4)	6을 입력한 다음 편집 커서를 쉼표 뒤로 이동하고 4를 입력합니다.
<b>ENTER</b>	nCr(6,4)	Enter 키를 눌러 결과를 표시합니다.
	15	

ALG 모드에서 사이 연산자는 **+**, **-**, **×**, **÷** 및 **y<sup>x</sup>**입니다. 다른 두 인수 연산에서는 f(x,y) 형태의 함수 표기법을 사용합니다. 여기서 x와 y는 순서대로 첫 번째 피연산자와 두 번째 피연산자입니다. RPN 모드에서 두 인수 연산의 피연산자는 Y, X 순으로 스택에 입력됩니다. 즉, y는 y 레지스터의 값이고 x는 x 레지스터의 값입니다.

이 규칙에서 y의 x 제곱근( $\sqrt[x]{y}$ )은 예외입니다. 예를 들어, RPN 모드에서  $\sqrt[3]{8}$ 을 계산하려면 **8** **ENTER** **3** **↵**  $\sqrt[x]{y}$ 를 누릅니다. ALG 모드에서 이와 동등한 연산은 **↵**  $\sqrt[x]{y}$  **3** **>** **8** **ENTER**로 입력됩니다.

단일 인수 연산과 마찬가지로 RPN 모드에서 두 인수 연산 중 일부는 ALG 모드와는 다르게 표시됩니다. 이러한 차이점은 아래 표에 요약되어 있습니다.

## 1-20 시작

키	RPN, RPN 프로그램에서	ALG, 방정식, ALG 프로그램에서
$y^x$	$y^x$	$^$
$x \overline{y}$	$x \overline{y}$	XROOT(, )
INT $\div$	INT $\div$	IDIV(, )

$\oplus$  및  $\otimes$ 와 같은 가환 연산의 경우 피연산자 순서가 계산된 결과에 영향을 주지 않습니다. RPN 모드에서 비가환인 두 인수 연산의 피연산자를 잘못된 순서로 입력한 경우  $\overleftrightarrow{xy}$  키를 눌러  $x$  레지스터와  $y$  레지스터의 내용을 교환합니다. 이 내용에 대해서는 2장에서 자세히 설명합니다(스택의  $X$  레지스터 및  $Y$  레지스터 교환절 참조).

## 디스플레이 형식 제어

모든 숫자는 12자리의 전체 자릿수로 저장됩니다. 그러나 디스플레이 메뉴의 옵션을 통해 숫자 표시에 사용되는 자릿수를 제어할 수 있습니다. 이 메뉴에 액세스하려면  $\leftarrow$  **DISPLAY**를 누릅니다. 처음 네 옵션(FIX, SCI, ENG 및 ALL)은 숫자 표시의 자릿수를 제어합니다. 일부 복잡한 내부 계산 중에 계산기는 중간 결과에 대해 15자리 전체 자릿수를 사용합니다. 표시된 숫자는 디스플레이 형식에 따라 *변* *올*림됩니다.





### 고정 소수 형식(FIX)

FIX 형식은 최대 소수 11자릿수(기수 표시 “.” 또는 “,” 오른쪽에 11자리)의 숫자(들어가는 경우)를 표시합니다. 프롬프트 FIX\_ 뒤에 표시할 소수 자릿수를 입력합니다. 10 또는 11 자릿수의 경우  $\leftarrow$  **0** 또는  $\leftarrow$  **1**을 누릅니다.

예를 들어, 숫자 123,456.7089에서 계산기를 FIX 4 디스플레이 모드로 설정할 때 표시되는 소수 자릿수는 “7”, “0”, “8” 및 “9”입니다.

너무 크거나( $10^{11}$ ) 너무 작아( $10^{-11}$ ) 현재 소수 자릿수 설정으로 표시할 수 없는 숫자는 자동으로 과학 형식으로 표시됩니다.

## 과학 형식(SCI)





SCI 형식은 최대 소수 11자릿수와 최대 3자릿수의 지수로 이루어진 유효숫자 표기법(기수 표시 “.” 또는 “,” 앞의 1자리)으로 숫자를 표시합니다. 프롬프트 SCI\_ 뒤에 표시할 소수 자릿수를 입력합니다. 10 또는 11 자릿수의 경우   또는   을 누릅니다. 숫자의 가수부는 항상 10보다 작습니다.

예를 들어, 숫자  $1.2346E5$ 에서 계산기를 SCI 4 디스플레이 모드로 설정할 때 표시되는 소수 자릿수는 “2”, “3”, “4” 및 “6”입니다. “E” 뒤의 “5”는 10의 지수입니다( $1.2346 \times 10^5$ ).





12자릿수보다 큰 숫자를 입력하거나 계산하는 경우 추가 전체 자릿수가 유지되지 않습니다.

## 공학 형식(ENG)






ENG 형식은 지수가 3의 배수인 점을 제외하고 유효숫자 표기법과 유사한 방식으로 숫자를 표시합니다. 기수 표시 “.” 또는 “,” 앞에 최대 3자릿수가 있을 수 있습니다. 이 형식은  $10^3$ 의 배수로 지정된 단위(예: 마이크로, 밀리 및 킬로 단위)를 사용하는 과학 및 공학 계산에 가장 유용합니다.



















프롬프트 ENG\_ 뒤에 첫 번째 유효 자릿수 뒤의 자릿수를 입력합니다. 10 또는 11 자릿수의 경우   또는   을 누릅니다.

예를 들어, 숫자  $123.46E3$ 에서 계산기를 ENG 4 디스플레이 모드로 설정할 때 표시되는 첫 번째 유효 자릿수 뒤의 유효 자릿수는 “2”, “3”, “4” 및 “6”입니다. “E” 뒤의 “3”은 10의 (3의 배수) 지수입니다( $123.46 \times 10^3$ ).

  또는   를 누르면 표시 중인 숫자의 지수 디스플레이가 3의 배수로 변경되고 가수가 적절하게 조정됩니다.

## 예:

이 예에서는 숫자 12.346E4를 사용하여 공학 형식의 동작을 보여 줍니다. 또한   ENG 및   ENG  함수의 사용을 보여 줍니다. 이 예에서는 RPN 모드를 사용합니다.

키:	디스플레이:	설명:
 DISPLAY 	ENG_	공학 형식을 선택합니다.
(3)ENG 	0.0000E0 0.0000E0	첫 번째 자릿수 뒤의 4 유효 자릿수를 나타내는 4를 입력합니다.
  .   	123.46E3	12.346E4를 입력합니다.
  ENTER	123.46E3	
  ENG 또는	123.46E3	지수를 3만큼 늘립니다.
 ENG 	123.46E3	
  ENG	123.46E3	
 ENG 	0.12346E6 123.46E3 123.46E3	지수를 3만큼 줄입니다.

## ALL 형식(ALL)

ALL 형식은 기본 디스플레이 형식으로, 숫자를 최대 12자리의 전체 자릿수로 표시합니다. 모든 자릿수가 디스플레이에 들어가지 않는 경우 숫자가 자동으로 과학 형식으로 표시됩니다.

## 숫자의 마침표 및 쉼표(·) (,)

HP 35s는 마침표와 쉼표를 사용하여 숫자를 보다 읽기 쉽도록 합니다. 마침표나 쉼표를 소수점(기수)으로 선택할 수 있습니다. 또한 천 단위 구분 기호를 사용하여 숫자를 3의 그룹으로 구분할지 여부를 선택할 수 있습니다. 다음 예에서는 옵션을 보여 줍니다.

## 예

숫자 12,345,678.90을 입력하고 소수점을 쉼표로 변경합니다. 그런 다음 천 단위 구분 기호가 없도록 선택합니다. 마지막으로, 기본 설정으로 돌아갑니다. 이 예에서는 RPN 모드를 사용합니다.

키:	디스플레이:	설명:
DISPLAY 4		
(4ALL)		
1 2 3 4 5	12,345,678.9	기본 형식에서 쉼표를 천 단위 구분 기호로, 마침표를 기수로 사용합니다.
6 7 8 . 9	12,345,678.9	
ENTER		
DISPLAY 6 (6.)	12.345.678.9	기수에 쉼표를 사용하도록 변경합니다. 천 단위 구분 기호는 자동으로 마침표로 변경됩니다.
	12.345.678.9	
DISPLAY 8	12345678.9	쉼표 구분 기호가 없도록 변경합니다.
(81000)	12345678.9	
DISPLAY 5 (5.)	12,345,678.9	기본 형식으로 돌아갑니다.
DISPLAY 7	12,345,678.9	
(71,000)		

## 복소수 디스플레이 형식( $x+iy$ , $x+yi$ , $r\theta$ )

복소수는  $x+iy$ ,  $x+yi$  및  $r\theta$ 와 같은 다양한 형식으로 표시할 수 있습니다. 단,  $x+yi$ 는 ALG 모드에서만 사용할 수 있습니다. 아래 예에서 복소수  $3+4i$ 는 세 가지 방식으로 모두 표시됩니다.





**DISPLAY** (4ALL) 58.5

모든 유효 자릿수이며 후행 0은 삭제됩니다.

**DISPLAY** (1FIX) 58.5000

소수 네 자릿수이며 지수는 없습니다.

0.0171

58.5의 역수입니다.

**SHOW** (누르고 있음) 170940170940

**SHOW** 를 놓을 때까지 전체 자릿수를 표시합니다.

---

## 분수

HP 35s를 사용하면 분수를 입력하고 연산하여 소수나 분수로 표시할 수 있습니다. HP 35s는  $a\ b/c$  형태로 분수를 표시합니다. 여기서  $a$ 는 정수이고  $b$ 와  $c$ 는 계산하는 숫자입니다. 또한  $b$ 는  $0 < b < c$ 이고  $c$ 는  $1 < c \leq 4095$ 로 설정됩니다.

### 분수 입력

언제든지 스택에 분수를 입력할 수 있습니다.

1. 숫자의 정수부를 입력하고 을 누릅니다. 첫 번째 은 숫자의 정수부를 소수부와 구분합니다.
2. 분수 분자를 입력하고 다시 을 누릅니다. 두 번째 은 분자를 분모와 구분합니다.
3. 분모를 입력한 다음 **ENTER** 또는 기능 키를 눌러 숫자 입력을 종료합니다. 숫자 또는 결과는 현재 디스플레이 형식에 따라 형식이 지정됩니다.

키 아래의  $a\ b/c$  기호는 키가 분수 입력에 두 번 사용됨을 나타냅니다.

다음 예에서는 분수 입력과 표시를 보여 줍니다.

## 예

혼합 숫자 12 3/8을 입력하고 분수 및 소수 형태로 표시합니다. 그런 다음 3/4을 입력하고 12 3/8에 더합니다. 이 예에서는 RPN 모드를 사용합니다.

키:	디스플레이:	설명:
<b>1</b> <b>2</b> <b>.</b> <b>3</b>	0 12.3	소수점은 정상적인 방법으로 해석됩니다.
<b>.</b> <b>8</b>	0.0000 12 3/8_	<b>.</b> 을 두 번째로 누르면 디스플레이가 분수 모드로 전환됩니다.
<b>ENTER</b>	12.3750 12.3750	입력 시 숫자가 현재 디스플레이 형식을 사용하여 표시됩니다.
<b>↵</b> <b>FDISP</b>	12 3/8 12 3/8	분수 디스플레이 모드로 전환합니다.
<b>.</b> <b>3</b> <b>.</b> <b>4</b>	12 3/8 0 3/4_	3/4을 입력합니다. 정수부가 없으므로 <b>.</b> 으로 시작합니다(0 3/4을 입력할 수 있음).
<b>+</b>	0 13 1/8	12 3/8에 3/4을 더합니다.
<b>↵</b> <b>FDISP</b>	0 13.1250	현재 디스플레이 모드로 다시 전환합니다.

분수 사용에 대한 자세한 내용은 5장, “분수”를 참조하십시오.



---

## 메시지

계산기는 ▲ 표시기를 표시하여 오류 상태에 응답합니다. 일반적으로 메시지와 함께 오류 표시기가 나타납니다.

- 메시지를 지우려면 **C** 또는 **←**를 누릅니다. RPN 모드에서는 오류 전 상태의 스택으로 돌아갑니다. ALG 모드에서는 마지막 수식으로 돌아가며, 오류를 수정할 수 있도록 편집 커서가 오류 위치에 표시됩니다.

- 아무 키나 눌러도 메시지가 지워지지만 키 기능은 입력되지 않습니다.

메시지가 표시되지 않지만 ▲ 표시기가 나타나는 경우 비활성 키나 잘못된 키를 눌렀습니다. 예를 들어, 이 컨텍스트에서는 두 번째 소수점에 의미가 없으므로  을 누르면 ▲가 표시됩니다.


표시되는 모든 메시지에 대해서는 부록 F, “메시지”에서 설명합니다.

---

## 계산기 메모리

HP 35s에는 모든 데이터 조합(변수, 방정식 또는 프로그램 줄)을 저장할 수 있는 30KB 메모리가 있습니다.

### 사용 가능한 메모리 확인

 **MEM**을 누르면 다음 메뉴가 표시됩니다.




```
1VAR  2 PGM
nnn   mm,mmm
```

여기서 각 항목은 다음을 나타냅니다.

nnn은 사용된 간접 변수의 양입니다.

mm,mmm은 사용 가능한 메모리의 바이트 수입니다.

**1** (1VAR)을 누르면 직접 변수 카탈로그가 표시됩니다(3장 “VAR 카탈로그의 변수 검토” 참조). **2** (2PGM)를 누르면 프로그램 카탈로그가 표시됩니다.

1. 변수 카탈로그를 시작하려면 **1** (1VAR)을 누르고, 프로그램 카탈로그를 시작하려면 **2** (2PGM)를 누릅니다.
2. 카탈로그를 검토하려면  또는 를 누릅니다.
3. 변수 또는 프로그램을 삭제하려면 해당 카탈로그에서 보는 동안  **CLEAR**를 누릅니다.
4. 카탈로그를 종료하려면 **C**를 누릅니다.

## 모든 메모리 지우기

모든 메모리를 지우면 저장한 숫자, 방정식 및 프로그램이 모두 지워집니다. 모드 및 형식 설정에는 영향을 주지 않습니다. 설정과 데이터를 지우려면 부록 B의 “메모리 지우기”를 참조하십시오.

### 모든 메모리를 지우려면

1. **4** (**4ALL**)를 누릅니다. 확인 프롬프트 **CLR ALL? Y NO**이 표시되어 실수로 메모리를 지우지 않도록 보호합니다.
2. **<** (**Y**) **ENTER**를 누릅니다.



## RPN: 자동 메모리 스택

---

이 장에서는 RPN 모드의 자동 메모리 스택에서 계산이 수행되는 방법에 대해 설명합니다. *계산기 사용을 위해 이 자료를 반드시 읽고 이해할 필요는 없지만*, 자료를 이해하면 특히 프로그래밍할 때 계산기를 훨씬 효과적으로 사용할 수 있습니다.

2부, “프로그래밍”에서는 프로그램 데이터를 조작하고 구성하는 데 스택이 어떻게 도움이 되는지를 알아보겠습니다.

---

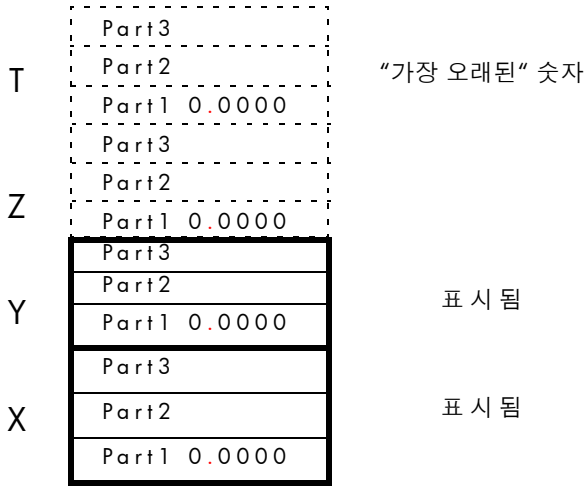
### 스택 정의

HP 35s에서 괄호를 사용하지 않고 복잡한 계산을 쉽게 처리할 수 있는 것은 *중간 결과가 자동으로 저장되기* 때문입니다. 자동 저장의 핵심은 *자동 RPN 메모리 스택*입니다.

HP의 연산 논리는 *괄호 없이* 사용되는 명백한 수학적 논리를 기반으로 합니다. 이 논리는 폴란드 논리학자 Jan Łukasiewicz(1878–1956)에 의해 개발되었으며 “폴란드 표기법”이라고 합니다.

기존의 대수 표기법에서는 관련 숫자나 변수 *사이*에 연산자가 들어갔지만 Łukasiewicz 표기법에서는 숫자나 변수 *앞에* 연산자가 들어갑니다. HP는 스택 효율성을 최적으로 높이기 위해 숫자 *뒤에* 연산자를 지정하도록 이 표기법을 수정했습니다. 따라서 이것을 RPN(*역 폴란드 표기법*)이라고 합니다.

스택은 서로 “누적”되는 *레지스터*라는 4개의 저장소 위치로 구성됩니다. X, Y, Z 및 T라는 레이블이 지정된 이러한 레지스터는 4개의 현재 숫자를 저장하고 조작합니다. “가장 오래된” 숫자가 T(*최상위*) 레지스터에 저장됩니다. 스택은 계산을 위한 작업 영역입니다.



가장 "최근" 숫자는 X 레지스터에 들어가며, *디스플레이의 두 번째 줄에 표시되는 숫자입니다.*

각 레지스터는 다음 세 부분으로 나누어져 있습니다.

- 실수 또는 1D 벡터가 part1을 사용하며 이 경우 part2와 part3은 null입니다.
- 복소수 또는 2D 벡터가 part1과 part2를 사용하며 이 경우 part3은 null입니다.
- 3D 벡터가 part1, part2 및 part3을 사용합니다.


프로그래밍 시 스택은 계산을 수행하고, 중간 결과를 임시로 저장하고, 저장된 데이터(변수)를 프로그램과 서브루틴 간에 전달하고, 입력을 받아들이며, 출력을 제공하는 데 사용됩니다.

## 2-2 RPN: 자동 메모리 스택










## 디스플레이 내의 X 레지스터와 Y 레지스터

메뉴, 메시지, 방정식 줄 또는 프로그램 줄이 표시되는 경우를 제외하고 사용자에게 표시되는 내용이 X 레지스터와 Y 레지스터입니다. 몇몇 기능 이름에도  $x$  또는  $y$ 가 포함되어 있는 것을 볼 수 있습니다.


이것은 우연의 일치가 아닙니다. 이 두 문자는 X 레지스터와 Y 레지스터를 나타냅니다. 예를 들어,   $10^x$ 은 10을 X 레지스터에 있는 숫자로 거듭제곱합니다.






## X 레지스터 지우기





 **CLEAR**  (\*)를 누르면 항상 X 레지스터가 0으로 지워지며, 이 명령을 프로그래밍하는 데에도 사용됩니다. 반면에 **C** 키는 상황에 맞게 사용됩니다. 상황에 따라 현재 디스플레이를 지우거나 취소하며, X 레지스터가 표시되는 경우에만  **CLEAR**  (\*)처럼 동작합니다. X 레지스터가 표시되고 숫자 입력이 종료된 경우(커서가 없음)  도  **CLEAR**  (\*)처럼 동작합니다.

## 스택 검토

### R↓(롤다운)



 (롤다운) 키를 사용하면 한 번에 한 레지스터씩, 내용을 아래로 “롤링”하여 스택의 전체 내용을 검토할 수 있습니다.  $x$  레지스터와  $y$  레지스터를 롤링할 때 숫자를 볼 수 있습니다.

스택이 1, 2, 3, 4로 채워져 있는 경우( **ENTER**  **ENTER**  **ENTER**  **ENTER**)  을 4회 누르면 숫자가 한 바퀴 회전하여 처음과 동일한 상태가 됩니다.

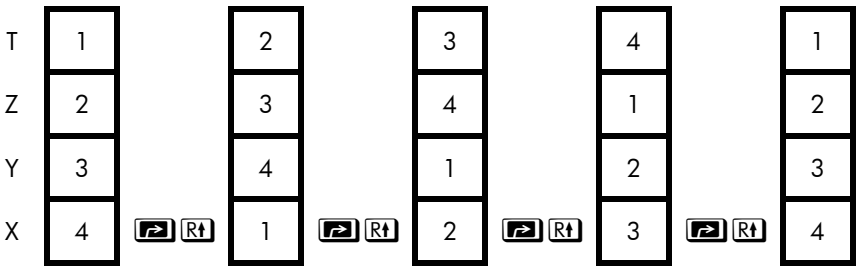
T	1		4		3		2		1
Z	2		1		4		3		2
Y	3		2		1		4		3
X	4		3		2		1		4

X 레지스터의 내용은 T 레지스터로 순환되고, T 레지스터의 내용은 Z 레지스터로 순환됩니다. 레지스터의 *내용만* 롤링되므로 레지스터 자체의 위치는 유지되고 X 레지스터와 Y 레지스터의 내용만 표시됩니다.



### R↑(롤업)


 **R↑** (롤업) 키는 한 번에 한 레지스터씩, 스택 내용을 위로 “롤링”한다는 점을 제외하고는  과 유사하게 작동합니다.

X 레지스터의 내용은 Y 레지스터로 순환되고, T 레지스터의 내용은 X 레지스터로 순환됩니다.



### 스택에서 X 레지스터와 Y 레지스터 교환

스택 내용을 조작하는 또 다른 키는  (*x와 y 교환*)입니다. 이 키는 스택의 나머지 부분에 영향을 주지 않으면서 X 레지스터와 Y 레지스터의 내용을 바꿉니다.  을 두 번 누르면 X 및 Y 레지스터 내용이 원래 순서로 복원됩니다.

 기능은 주로 계산에서 숫자의 순서를 바꾸는 데 사용됩니다.

예를 들어,  $9 \div (13 \times 8)$ 을 계산하려면

        를 누릅니다.

이 수식을 *왼쪽에서 오른쪽으로* 계산하는 키 입력은

        입니다.

#### 참고



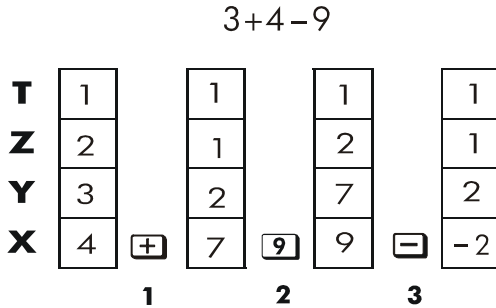
스택에 있는 숫자의 개수는 항상 4개를 넘지 않습니다.

5번째 숫자를 입력하면 T 레지스터(최상위 레지스터)의 내용은 손실됩니다.

## 산술 - 스택의 산술 수행 방법

새로운 숫자가 X 레지스터에 입력되어 *스택을 위로 올리고* 연산자가 X 및 Y 레지스터의 두 숫자를 결합한 결과 X 레지스터에 새로운 숫자가 하나 생성되어 *스택을 아래로 내리면* 스택의 내용이 위/아래로 자동으로 이동합니다.

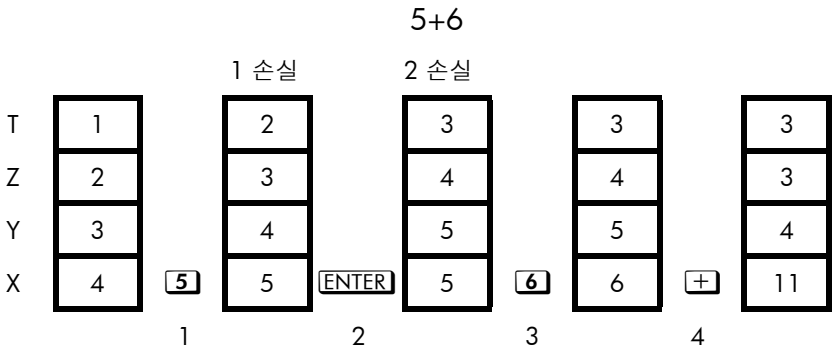
스택에 숫자 1, 2, 3 및 4가 채워져 있는 경우 계산 중에 스택의 내용이 어떻게 내려가고 올라가는지 확인합니다.



1. 스택의 내용이 “내려갑니다”. T(최상위) 레지스터의 내용이 복제됩니다.
2. 스택의 내용이 “올라갑니다”. T 레지스터의 내용이 손실됩니다.
3. 스택이 내려갑니다.
  - 스택이 올라가면 T(최상위) 레지스터의 내용이 Z 레지스터의 내용으로 대체 되고 T 레지스터의 *이전* 내용은 손실됩니다. 따라서 스택의 메모리가 4개 숫자로 제한되는 것을 확인할 수 있습니다.
  - 스택은 자동으로 이동하므로 새로운 계산을 수행하기 전에 X 레지스터를 지우지 *않아도* 됩니다.
  - 대부분의 기능은 *다음 숫자가 X 레지스터에 들어올 때 스택의 내용을 올릴 준비를 합니다.* 스택 올리기를 사용하지 않는 기능 목록은 부록 B를 참조하십시오.

## ENTER 작동 방식

이미 알고 있듯이 **ENTER** 는 나란히 입력된 두 숫자를 분리합니다. 그러면 스택에서는 이 기능이 어떠한 동작을 하는지 알아보겠습니다. 스택이 다시 1, 2, 3, 4로 채워져 있다고 가정하고, 이제 두 개의 새로운 숫자를 입력하여 더합니다.



1. 스택을 올립니다.
2. 스택을 올리고 X 레지스터를 복제합니다.
3. 스택을 올리지 *않습니다*.
4. 스택을 내리고 X 레지스터를 복제합니다.

**ENTER** 는 X 레지스터의 내용을 Y 레지스터로 복제합니다. 다음에 입력하거나 다시 호출하는 숫자는 X 레지스터에 남아 있는 첫 번째 숫자의 복사본을 *덮어씁니다*. 그 결과 순차적으로 입력된 두 개의 숫자가 분리됩니다.

**ENTER** 의 복제 기능을 사용하여 스택을 신속하게 지울 수 있습니다.

0 **ENTER** **ENTER** **ENTER** 를 누릅니다. 이제 모든 스택 레지스터에 0이 됩니다. 그러나 계산을 수행하기 전에 스택을 지울 필요는 *없습니다*.

### 한 숫자를 연속으로 두 번 사용

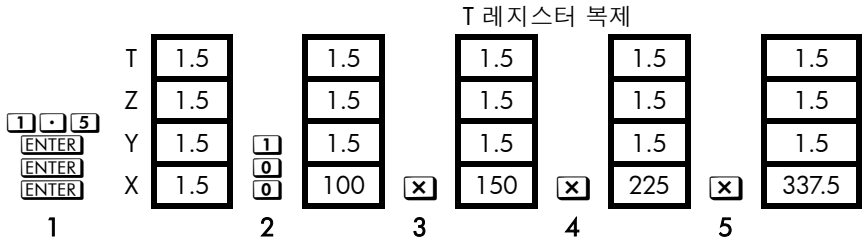
**ENTER** 의 복제 기능을 다른 방식으로 사용할 수도 있습니다. 숫자를 해당 숫자와 더하려면 **ENTER** **+** 를 누릅니다.

## 스택에 상수 채우기

[ENTER]의 복제 기능과 스택 내리기의 복제 기능(T에서 Z 방향)을 함께 사용하면 스택을 계산에 사용할 숫자 상수로 채울 수 있습니다.

예:

매일 50%씩 일정하게 성장하는 세균 배양에서 100개의 개체 수는 3일 후에 얼마나 증가하겠습니까?



1. 스택에 성장률을 채웁니다.
2. 초기 개체 수를 입력합니다.
3. 1일 후의 개체 수를 계산합니다.
4. 2일 후의 개체 수를 계산합니다.
5. 3일 후의 개체 수를 계산합니다.

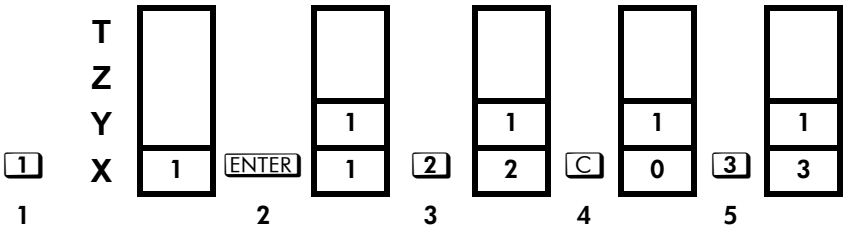
## 스택을 지우는 방법

X 레지스터를 지우면 0이 X 레지스터에 들어갑니다. 다음에 입력하거나 다시 호출하는 숫자가 이 0을 덮어씁니다.

X 레지스터의 내용을 지우는, 즉 x를 지우는 방법에는 다음 네 가지가 있습니다.

1. C를 누릅니다.
2. ←를 누릅니다.
3. ▶ CLEAR 1(1x)를 누릅니다. 주로 프로그램 입력 중에 사용됩니다.
4. ▶ CLEAR 5(5STK)를 눌러 X, Y, Z 및 T 레지스터를 0으로 지웁니다.

예를 들어, 1과 3을 입력하려고 했는데 실수로 1과 2를 입력한 경우 오류를 수정하려면 다음을 수행해야 합니다.



1. 스택을 올립니다.
2. 스택을 올리고 X 레지스터를 복제합니다.
3. X 레지스터를 덮어씁니다.
4. 0으로 덮어써서  $x$ 를 지웁니다.
5.  $x$ 를 덮어씁니다(0으로 대체).

## LAST X 레지스터

LAST X 레지스터는 스택의 부속 항목으로 마지막 숫자 기능이 실행되기 전에 X 레지스터에 있었던 숫자를 저장합니다. 숫자 기능은 다른 숫자에서 값을 구하는 연산(예:  $\sqrt{x}$ )입니다.  $\rightarrow$  [LAST  $x$ ]를 누르면 이 값이 X 레지스터로 반환됩니다.

“마지막  $x$ ”를 검색하는 이 기능의 두 가지 주요 용도는 다음과 같습니다.

1. 오류를 수정합니다.
2. 계산에 숫자를 다시 사용합니다.

$x$ 를 LAST X 레지스터에 저장하는 기능의 전체 목록은 부록 B를 참조하십시오.

# LAST X를 사용하여 오류 수정

## 잘못된 단일 인수 기능

잘못된 단일 인수 기능을 실행하는 경우 올바른 기능을 실행할 수 있도록 **[F2] [LAST X]**를 사용하여 숫자를 검색합니다. 잘못된 결과를 스택에서 지우려는 경우 **[C]**를 먼저 누릅니다.

**[F2] [%]** 및 **[F2] [%CHG]**에서는 스택이 내려가지 않으므로 단일 인수 기능과 동일한 방식으로 이러한 기능에서 복구할 수 있습니다.

### 예:

방금  $\ln 4.7839 \times (3.879 \times 10^5)$ 을 계산했으며 제곱근을 구하려고 했는데 실수로 **[e<sup>x</sup>]**을 누른 경우 처음부터 계산을 다시 시작하지 않아도 됩니다. 올바른 결과를 구하려면 **[F2] [LAST X] [√x]**를 누릅니다.

## 두 인수 기능의 오류

두 인수 연산(예: **[√x]**, **[y<sup>x</sup>]** 또는 **[lnCr]**)에서 실수를 한 경우 **[F2] [LAST X]** 및 두 인수 연산의 역을 사용하여 수정할 수 있습니다.

1. **[F2] [LAST X]**를 눌러 두 번째 숫자(연산 직전의  $x$ )를 복구합니다.
2. 역연산을 실행합니다. 원래 첫 번째 숫자였던 숫자가 반환됩니다. 두 번째 숫자는 여전히 LAST X 레지스터에 있습니다. 그런 후 다음을 수행합니다.
  - 잘못된 기능을 사용한 경우 **[F2] [LAST X]**를 다시 눌러 원래 스택 내용을 복원합니다. 이제 올바른 기능을 실행합니다.
  - 잘못된 두 번째 숫자를 사용한 경우 올바른 숫자를 입력하고 기능을 실행합니다.

잘못된 첫 번째 숫자를 사용한 경우 올바른 첫 번째 숫자를 입력하고 **[F2] [LAST X]**를 눌러 두 번째 숫자를 복구한 다음 기능을 다시 실행합니다. 잘못된 결과를 스택에서 지우려는 경우 **[C]**를 먼저 누릅니다.

예:

다음을 계산하는 동안 실수를 했다고 가정합니다.

$$16 \times 19 = 304$$

다음과 같은 세 종류의 실수를 했을 수 있습니다.

**잘못된 계산:**

**1** **6** **ENTER** **1**

**9** **=**

**1** **5** **ENTER** **1**

**9** **×**

**1** **6** **ENTER** **1**

**8** **×**

**실수:**

잘못된 기능

잘못된 첫 번째 숫자

잘못된 두 번째 숫자

**수정:**

**↶** **LASTx** **+**

**↶** **LASTx** **×**

**1** **6** **↶** **LASTx** **×**

**↶** **LASTx** **÷** **1** **9** **×**

## LAST X를 사용하여 숫자 다시 사용

**↶** **LASTx** 를 사용하여 숫자(예: 상수)를 계산에 다시 사용할 수 있습니다.

이때 상수가 X 레지스터에서 마지막 숫자가 되어 **↶** **LASTx** 를 통해 저장 및 검색될 수 있도록 산술 연산을 실행하기 직전에 상수를 두 번 입력해야 합니다.

예:

$$\frac{96.704 + 52.3947}{52.3947} \text{ 계산}$$



	T	<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>		<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>		<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>
<i>t</i>									
<i>t</i>									
<i>t</i>									
<b>9</b> <b>6</b>	Z	<table border="1"><tr><td><i>z</i></td></tr></table>	<i>z</i>	<b>5</b> <b>2</b>	<table border="1"><tr><td><i>z</i></td></tr></table>	<i>z</i>		<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>
<i>z</i>									
<i>z</i>									
<i>t</i>									
<b>.</b> <b>7</b>	Y	<table border="1"><tr><td>96.7040</td></tr></table>	96.7040	<b>.</b> <b>3</b>	<table border="1"><tr><td>96.7040</td></tr></table>	96.7040		<table border="1"><tr><td><i>z</i></td></tr></table>	<i>z</i>
96.7040									
96.7040									
<i>z</i>									
<b>0</b> <b>4</b>	X	<table border="1"><tr><td>96.7040</td></tr></table>	96.7040	<b>9</b> <b>4</b>	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947	<b>+</b>	<table border="1"><tr><td>149.0987</td></tr></table>	149.0987
96.7040									
52.3947									
149.0987									
<b>ENTER</b>			<b>7</b>						

LAST	X	<table border="1"><tr><td>/</td></tr></table>	/		<table border="1"><tr><td>/</td></tr></table>	/	<b>+</b>	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947
/									
/									
52.3947									

	T	<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>		<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>
<i>t</i>						
<i>t</i>						
	Z	<table border="1"><tr><td><i>z</i></td></tr></table>	<i>z</i>		<table border="1"><tr><td><i>t</i></td></tr></table>	<i>t</i>
<i>z</i>						
<i>t</i>						
	Y	<table border="1"><tr><td>149.0987</td></tr></table>	149.0987		<table border="1"><tr><td><i>z</i></td></tr></table>	<i>z</i>
149.0987						
<i>z</i>						
<b>↶</b> <b>LASTx</b>	X	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947	<b>÷</b>	<table border="1"><tr><td>2.8457</td></tr></table>	2.8457
52.3947						
2.8457						

LAST	X	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947		<table border="1"><tr><td>52.3947</td></tr></table>	52.3947
52.3947						
52.3947						

**키:**

**디스플레이:**

**설명:**

**9** **6** **.** **7** **0** **4**  
**ENTER**

96.7040

첫 번째 숫자를 입력합니다.

**5** **2** **.** **3** **9** **4**

149.0987

중간 결과입니다.

**7** **+**

52.3947

**+** 앞에서 디스플레이를 가져옵니다.

**↶** **LASTx**

2.8457

최종 결과입니다.

**÷**

**예:**

지구에서 가까운 두 행성은 리겔 센타우루스(4.3광년 거리)와 시리우스(8.7광년 거리)입니다. 광속( $9.5 \times 10^{15}$ 미터/년) *c*를 사용하여 지구에서 두 별까지의 거리를 미터로 변환합니다.

리겔 센타우루스까지의 거리:  $4.3 \text{ yr} \times (9.5 \times 10^{15} \text{ m/yr})$ .

시리우스까지의 거리:  $8.7 \text{ yr} \times (9.5 \times 10^{15} \text{ m/yr})$ .

키:	디스플레이:	설명:
<b>4</b> <b>.</b> <b>3</b> <b>[ENTER]</b>	4.3000	리겔 센타우루스까지의 광년입니다.
<b>9</b> <b>.</b> <b>5</b> <b>[E]</b> <b>1</b> <b>5</b>	9.5E15_	광속 c입니다.
<b>[X]</b>	4.0850E16	리겔 센타우루스까지의 미터입니다.
<b>8</b> <b>.</b> <b>7</b> <b>[&gt;]</b> <b>[LASTx]</b>	9.5000E15	c를 검색합니다.
<b>[X]</b>	8.2650E16	시리우스까지의 미터입니다.

## RPN 모드의 연속 계산

RPN 모드에서는 스택의 내용이 자동으로 올라가고 내려가 값을 저장 또는 다시 입력하거나 괄호를 사용하지 않고도 중간 결과를 유지할 수 있습니다.

### 괄호 없이 작업

예를 들어,  $(12 + 3) \times 7$ 을 계산합니다.

종이에서 이 문제를 계산하는 경우 먼저  $(12 + 3)$ 의 중간 결과를 계산합니다.

$$(12 + 3) = 15$$

그런 다음 중간 결과에 7을 곱합니다.

$$(15) \times 7 = 105$$

괄호 안에서 시작하여 HP 35s에서 동일한 방식으로 이 수식을 계산합니다.

키:	디스플레이:	설명:
<b>1</b> <b>2</b> <b>[ENTER]</b> <b>3</b> <b>[+]</b>	15.0000	먼저 중간 결과를 계산합니다.

계속하기 전에 이 중간 결과를 저장하기 위해 **[ENTER]**를 누르지 않아도 됩니다. 이 값은 계산 결과이기 때문에 자동으로 저장됩니다.

## 2-12 RPN: 자동 메모리 스택

<b>키:</b>	<b>디스플레이:</b>	<b>설명:</b>
<b>7</b> <b>X</b>	105.0000	기능 키를 누르면 결과가 계산됩니다. 이 결과를 이후 계산에 사용할 수 있습니다.

이제 다음 예를 살펴보겠습니다. 수식을 시작할 때와 같이 순차적으로 입력된 숫자를 분리하려는 경우에만 **ENTER**를 누르면 됩니다. **+**, **-** 등과 같은 연산 자체에서 이후 숫자를 분리하고 중간 결과를 저장합니다. 계산을 수행하는 데 필요한 경우 첫 번째로 검색되는 숫자는 마지막으로 저장된 결과입니다.

2 ÷ (3 + 10) 계산:

<b>키:</b>	<b>디스플레이:</b>	<b>설명:</b>
<b>3</b> <b>ENTER</b> <b>1</b> <b>0</b> <b>+</b>	13.0000	(3 + 10)을 먼저 계산합니다.
<b>2</b> <b>X↔Y</b> <b>÷</b>	0.1538	나누기가 올바르게 2를 13 <i>앞에</i> 놓습니다(2 ÷ 13).

4 ÷ [14 + (7 × 3) - 2] 계산:

<b>키:</b>	<b>디스플레이:</b>	<b>설명:</b>
<b>7</b> <b>ENTER</b> <b>3</b> <b>X</b>	21.0000	(7 × 3)을 계산합니다.
<b>1</b> <b>4</b> <b>+</b> <b>2</b> <b>-</b>	33.0000	분모를 계산합니다.
<b>4</b> <b>X↔Y</b>	33.0000	나누기를 위해 4를 33 <i>앞에</i> 놓습니다.
<b>÷</b>	0.1212	4 ÷ 33을 계산한 결과입니다.

중간 결과가 자동으로 저장되기 때문에 괄호가 여러 개 있는 문제도 동일한 방식으로 값을 구할 수 있습니다. 예를 들어, 종이에서 (3 + 4) × (5 + 6)의 값을 구하려면 먼저 수량 (3 + 4)를 계산합니다. 그런 다음 (5 + 6)을 계산합니다. 마지막으로 두 개의 중간 결과를 곱하여 결과를 구합니다.

중간 결과를 기록할 필요가 없다는 점을 제외하고 HP 35s에서도 동일한 방식으로 문제를 풀며 계산기에서 중간 결과를 저장합니다.

<b>키:</b>	<b>디스플레이:</b>	<b>설명:</b>
<b>3</b> <b>ENTER</b> <b>4</b> <b>+</b>	21.0000	먼저 (3+4)를 더합니다.
<b>5</b> <b>ENTER</b> <b>6</b> <b>+</b>	33.0000	그런 다음 (5+6)을 더합니다.
<b>X</b>	33.0000	중간 결과를 곱하여 최종 결과를 구합니다.

## 연습

계산:

$$\frac{\sqrt{(16.3805 \times 5)}}{0.05} = 181.0000$$

풀이:

$$\boxed{1} \boxed{6} \boxed{\cdot} \boxed{3} \boxed{8} \boxed{0} \boxed{5} \boxed{\text{ENTER}} \boxed{5} \boxed{\times} \boxed{\sqrt{x}} \boxed{\cdot} \boxed{0} \boxed{0} \boxed{5} \boxed{\div}$$

계산:

$$\sqrt{[(2+3) \times (4+5)]} + \sqrt{[(6+7) \times (8+9)]} = 21.5743$$

풀이:

$$\boxed{2} \boxed{\text{ENTER}} \boxed{3} \boxed{+} \boxed{4} \boxed{\text{ENTER}} \boxed{5} \boxed{+} \boxed{\times} \boxed{\sqrt{x}} \boxed{6} \boxed{\text{ENTER}} \boxed{7} \boxed{+} \boxed{8} \boxed{\text{ENTER}} \boxed{9} \boxed{+} \boxed{\times} \boxed{\sqrt{x}} \boxed{+}$$

계산:

$$(10 - 5) \div [(17 - 12) \times 4] = 0.2500$$

풀이:

$$\boxed{1} \boxed{7} \boxed{\text{ENTER}} \boxed{1} \boxed{2} \boxed{-} \boxed{4} \boxed{\times} \boxed{1} \boxed{0} \boxed{\text{ENTER}} \boxed{5} \boxed{-} \boxed{x \leftrightarrow y} \boxed{\div}$$

또는

$$\boxed{1} \boxed{0} \boxed{\text{ENTER}} \boxed{5} \boxed{-} \boxed{1} \boxed{7} \boxed{\text{ENTER}} \boxed{1} \boxed{2} \boxed{-} \boxed{4} \boxed{\times} \boxed{\div}$$

## 계산 순서

맨 안쪽 괄호에서 바깥쪽으로 연속 계산하는 것이 좋습니다. 그러나 왼쪽에서 오른쪽으로 문제를 풀 수도 있습니다.

예를 들어, 다음 수식을 이미 계산했습니다.

$$4 \div [14 + (7 \times 3) - 2]$$

이 경우 연필과 종이를 사용할 때와 마찬가지로 맨 안쪽 괄호(7 × 3)에서 시작하여 바깥쪽으로 계산했습니다. 키 입력은 **7** **ENTER** **3** **x** **1** **4** **+** **2** **-** **4** **x↔y** **÷**였습니다.

왼쪽에서 오른쪽으로 문제를 푸는 경우

**4** **ENTER** **1** **4** **ENTER** **7** **ENTER** **3** **x** **+** **2** **-** **÷**를 누릅니다.

이 방법을 사용하면 키 입력이 하나 더 추가됩니다. 첫 번째 중간 결과는 여전히 맨 안쪽 괄호(7 × 3)입니다. 왼쪽에서 오른쪽으로 문제를 푸는 경우 *비가환* 기능 (**-** 및 **÷**)의 피연산자 위치를 조정하기 위해 **x↔y**을 사용할 필요가 없다는 이점이 있습니다.

그러나 첫 번째 방법(맨 안쪽 괄호에서 시작)이 대체로 선호되는 것은 다음과 같은 이유 때문입니다.

- 키 입력이 더 적습니다.
- 스택에 필요한 레지스터 개수가 더 적습니다.

### 참고



*왼쪽에서 오른쪽* 방법을 사용하는 경우 한 번에 필요한 중간 숫자 또는 결과가 4개를 넘지 않도록 해야 합니다. 스택에는 최대 4개의 숫자만 저장될 수 있습니다.

위의 예를 *왼쪽에서 오른쪽*으로 계산했을 때는 한 시점에서 스택의 모든 레지스터가 필요했습니다.

키:	디스플레이:	설명:
<b>4</b> <b>ENTER</b> <b>1</b> <b>4</b>	14.0000	4와 14를 스택에 중간 결과로 저장합니다.
<b>ENTER</b>		
<b>7</b> <b>ENTER</b> <b>3</b>	3_	이때 스택이 이 계산을 위한 숫자로 꽉 차게 됩니다.
<b>x</b>	21.0000	중간 결과입니다.
<b>+</b>	35.0000	중간 결과입니다.
<b>2</b> <b>-</b>	33.0000	중간 결과입니다.
<b>÷</b>	0.1212	최종 결과입니다.

## 추가 연습

다음 문제를 풀어 RPN 사용을 연습합니다.

계산:

$$(14 + 12) \times (18 - 12) \div (9 - 7) = 78.0000$$

풀이:

**1 4** **ENTER** **1 2** **+** **1 8** **ENTER** **1 2** **-** **x** **9** **ENTER** **7** **-** **÷**

계산:

$$23^2 - (13 \times 9) + 1/7 = 412.1429$$

풀이:

**2 3** **→** **x<sup>2</sup>** **1 3** **ENTER** **9** **x** **-** **7** **1/x** **+**

계산:

$$\sqrt{(5.4 \times 0.8) \div (12.5 - 0.7^3)} = 0.5961$$

풀이:

**5** **.** **4** **ENTER** **.** **8** **x** **.** **7** **ENTER** **3** **y<sup>x</sup>** **1 2** **.** **5** **x↔y** **-**  
**÷** **√x**

또는

**5** **.** **4** **ENTER** **.** **8** **x** **1 2** **.** **5** **ENTER** **.** **7** **ENTER** **3** **y<sup>x</sup>** **-**  
**÷** **√x**

계산:

$$\sqrt{\frac{8.33 \times (4 - 5.2) \div [(8.33 - 7.46) \times 0.32]}{4.3 \times (3.15 - 2.75) - (1.71 \times 2.01)}} = 4.5728$$

풀이:

**4** **ENTER** **5** **.** **2** **-** **8** **.** **3** **3** **x** **→** **LASTx** **7** **.** **4** **6** **-**  
**0** **.** **3** **2** **x** **÷** **3** **.** **1** **5** **ENTER** **2** **.** **7** **5** **-** **4** **.** **3** **x**  
**1** **.** **7** **1** **ENTER** **2** **.** **0** **1** **x** **-** **÷** **√x**

## 변수에 데이터 저장

HP 35s에는 숫자, 방정식 및 프로그램을 저장할 수 있는 30KB 메모리가 있습니다. 숫자는 각각 A에서 Z까지의 문자로 이름이 지정된 변수라는 위치에 저장됩니다. (은행 잔액은 B, 광속은 C처럼 저장된 내용을 식별하기 쉬운 문자를 선택하면 됩니다.)









### 예:

이 예에서는 RPN 모드와 ALG 모드에서 값 3을 변수 A에 저장하는 방법을 차례로 보여 줍니다.

키:	디스플레이:	설명:
<b>MODE</b> <b>5</b> (5 RPN)		RPN 모드로 전환합니다(필요한 경우).
<b>3</b>	0.0000 3_	값(3)을 입력합니다.
<b>STO</b>	STO_	저장 명령에서 문자를 묻는 프롬프트를 표시합니다. A...Z 표시기를 확인합니다.
<b>A</b>	0.0000 3.0000	값 3이 A에 저장되고 스택에 반환됩니다.
<b>MODE</b> <b>4</b> (4 ALG)	3.0000	ALG 모드로 전환합니다(필요한 경우).
<b>3</b> <b>STO</b> <b>A</b>	3▶A_	다시 저장 명령에서 문자를 묻는 프롬프트를 표시하며 A...Z 표시기가 나타납니다.
<b>ENTER</b>	3▶A 3.0000	값 3이 A에 저장되고 결과가 줄 2에 표시됩니다.

ALG 모드에서 수식을 변수에 저장할 수 있습니다. 이 경우 수식 자체가 아니라 수식의 값이 변수에 저장됩니다.

예:




키:	디스플레이:	설명:
  	1+3÷4 	수식을 입력한 다음 이전 예와
   	1.7500	같이 진행합니다.

각 분홍색 문자는 키 및 고유한 변수와 연결되어 있습니다. 디스플레이의 **A..Z** 표시기에서 이를 확인합니다.

변수 *X*, *Y*, *Z* 및 *T*는 스택에 있는 X 레지스터, Y 레지스터, Z 레지스터 및 T 레지스터의 서로 다른 저장 위치입니다.

---

## 숫자 저장 및 다시 호출

숫자와 벡터는 저장( ) 및 다시 호출() 명령을 통해 문자가 지정된 변수에 저장되고 다시 호출됩니다. 숫자는 실수 또는 복소수, 소수 또는 분수, 10진수 또는 HP 35s에서 지원되는 기타 형식일 수 있습니다.

### 표시된 숫자(X 레지스터)의 복사본을 직접 변수에 저장하려면

  문자 키  를 누릅니다.

### 직접 변수의 숫자 복사본을 디스플레이로 다시 호출하려면

 문자 키  를 누릅니다.

### 예: 숫자 저장

아보가드로 수(약  $6.0221 \times 10^{23}$ )를 A에 저장합니다.



키:	디스플레이:	설명:
6 . 0 2 2 1	6.0221E23_	아보가드로 수입니다.
E 2 3	6.0221E23▶A_	“▶” 변수를 묻는 프롬프트를 표시합니다.
▶ STO A	6.0221E23▶A	아보가드로 수의 복사본을 A에 저장합니다. 또한 숫자 입력이 종료됩니다.
ENTER	6.0221E23	디스플레이의 숫자를 지웁니다.
C	-	A..Z 표시기가 켜집니다.
RCL	A..Z	A 디스플레이에서 아보가드로 수를 복사합니다.
A ENTER	A=	
	6.0221E23	

변수에 저장된 값을 다시 호출하려면 다시 호출 명령을 사용합니다. 이 명령의 디스플레이는 다음 예와 같이 RPN 모드와 ALG 모드에서 약간 다릅니다.

#### 예:

이 예에서는 마지막 예에서 변수 G에 저장한 값 1.75를 다시 호출합니다. 시작할 때는 HP 35s가 ALG 모드에 있다고 가정합니다.

키:	디스플레이:	설명:
RCL G ENTER	G	RCL을 누르면 단순히 A..Z 모드가 활성화되고 명령이 줄 1에 붙여넣어 지지 않습니다.
	1.7500	

ALG 모드에서 다시 호출을 사용하여 변수를 명령줄의 수식에 붙여넣을 수 있습니다. 위의 G=1.75를 사용하여 15·2×G를 계산한다고 가정합니다.

키:	디스플레이:	설명:
1 5 - 2 ×	15-2×G	
RCL G ENTER	11.5000	

이제 RPN 모드로 전환하고 G의 값을 다시 호출합니다.

**키:**  
MODE 5 (SRPN)

**디스플레이:**

**설명:**  
RPN 모드로 전환합니다.

RCL

G

RCL \_  
1.7500  
1.7500

RPN 모드에서 RCL은 명령을 편집  
줄에 붙여넣습니다.  
ENTER를 누를 필요가 없습니다.

---

## 변수 보기

VIEW 명령(← VIEW)은 변수 값을 X 레지스터로 다시 호출하지 않고 그 값을 표시합니다. 디스플레이는 변수=값의 형태로 표시됩니다. 숫자의 자릿수가 너무 많아 디스플레이에 모두 들어가지 않는 경우 → > 또는 → <를 사용하여 누락된 숫자를 봅니다. VIEW 디스플레이를 취소하려면 ← 또는 C를 누릅니다. VIEW 명령은 대체로 프로그래밍에서 사용되지만 언제든지 스택에 영향을 주지 않고 변수 값을 보려는 경우에 유용합니다.

---

## MEM 카탈로그 사용

MEMORY 카탈로그(← VIEW)에서는 사용 가능한 메모리 양에 대한 정보를 제공합니다. 카탈로그 디스플레이는 다음과 같은 형식으로 표시됩니다.

1. VAR    2. PGM

nnn        mm,mmm

여기서 mm,mmm은 사용 가능한 메모리의 바이트 수이고, nnn은 사용된 간접 변수의 양입니다.

간접 변수에 대한 자세한 내용은 14장을 참조하십시오.

## VAR 카탈로그

기본적으로 A에서 Z까지의 모든 직접 변수에는 값 0이 포함되어 있습니다. 직접 변수에 0이 아닌 값을 저장하면 VAR 카탈로그(← MEM 1 (1VAR))에서 해당 변수의 값을 볼 수 있습니다.

### 3-4 변수에 데이터 저장

**예:**

이 예에서는 C에 3, D에 4, E에 5를 각각 저장한 다음 VAR 카탈로그를 통해 이러한 변수를 보고 지웁니다. 이 예에서는 RPN 모드를 사용합니다.

키:	디스플레이:	설명:
<b>CLEAR</b> <b>2</b> (2VAR)		모든 직접 변수를 지웁니다.
<b>S</b> )		
<b>STO</b> <b>C</b>	4	C에 3, D에 4, E에 5를 각각 저장합니다.
<b>STO</b> <b>D</b>	5	
<b>STO</b> <b>E</b>		
<b>MEM</b> <b>1</b> (1VAR)	C=	VAR 카탈로그를 시작합니다.
	3	

↑ 및 ↓ 표시기는 카탈로그 스크롤에 도움이 되도록 및 키가 활성화되었음을 나타내기 위해 표시됩니다. 그러나 분수 표시 모드가 활성화 상태이면 카탈로그에 변수가 하나뿐인 경우를 제외하고 정확성을 나타내기 위해 ▲ 및 ▼ 표시기가 활성화되지 않습니다. 이제 VAR 카탈로그 탐색 방법을 보여 주는 예로 돌아가겠습니다.

	D=	값이 0이 아닌 다음 직접 변수(D=4)까지 아래로 스크롤합니다.
	4	
	E=	다시 아래로 스크롤하여 E=5를 확인합니다.
	5	

VAR 카탈로그에 있는 동안, 이 예에서 변수 값을 0으로 지워 현재 값을 효과적으로 삭제하는 방법을 살펴보겠습니다. E를 삭제합니다.

<b>CLEAR</b>	C=	값이 0이므로 E가 더 이상 VAR 카탈로그에 존재하지 않습니다. 다음 변수는 표시된 것처럼 C입니다.
	3	

이제 C의 값을 스택에 복사한다고 가정합니다.

<b>ENTER</b>	5	C=3의 값이 X 레지스터에 복사되고 이전의 E=5 정의에서 가져온 5는 Y 레지스터로 이동합니다.
	3	

언제든지 VAR 카탈로그를 닫으려면 **←ENG** 또는 **C**를 누릅니다. 변수를 지우는 다른 방법으로 간단하게 변수에 값 0을 저장할 수도 있습니다. 마지막으로 **↵** **CLEAR** **2** (**2VARS**)를 눌러 모든 직접 변수를 지울 수 있습니다. 모든 직접 변수의 값이 0인 경우 VAR 카탈로그를 시작하려고 하면 "ALL VARS = 0"이라는 오류 메시지가 표시됩니다.

변수 값의 자릿수가 너무 많아 완전히 표시할 수 없는 경우 **→** 또는 **←**를 사용하여 누락된 숫자를 볼 수 있습니다.

---

## ✓ 저장된 변수를 사용한 산술

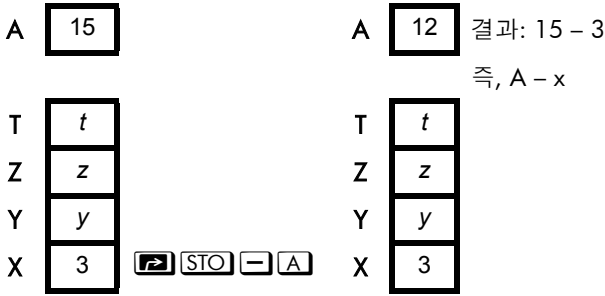
저장 산술 및 다시 호출 산술을 사용하면 변수를 스택으로 다시 호출하지 않고도 변수에 저장된 숫자를 사용하여 계산을 수행할 수 있습니다. 계산은 X 레지스터의 한 숫자와 지정된 변수의 한 숫자를 사용합니다.

## ✓ 저장 산술

저장 산술은 **↵** **STO** **+**, **↵** **STO** **-**, **↵** **STO** **x** 또는 **↵** **STO** **÷**를 사용하여 변수 자체에서 산술을 수행하고 결과를 해당 변수에 저장합니다. X 레지스터의 값이 사용되며 스택에는 영향을 주지 않습니다.

새로운 변수 값 = 이전 변수 값 {+, -, ×, ÷} x

예를 들어, A(15)의 값을 X 레지스터의 숫자(표시된 3)만큼 차감한다고 가정합니다. **↵** **STO** **-** **A**를 누릅니다. 이제 A = 12이고 디스플레이에는 여전히 3이 표시됩니다.

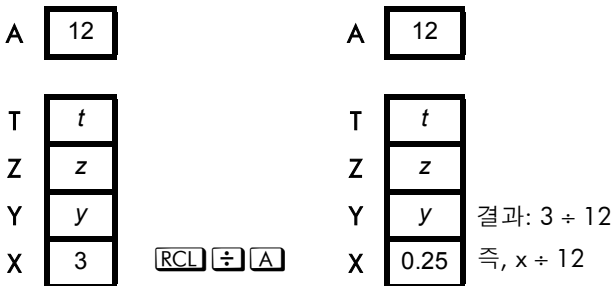


## ✓ 다시 호출 산술

다시 호출 산술은 RCL +, RCL -, RCL x 또는 RCL ÷를 사용하여 X 레지스터에서 다시 호출된 숫자로 산술을 수행하고 결과를 디스플레이에 유지합니다. X 레지스터만 영향을 받습니다. 변수의 값은 동일하게 유지되고 결과가 X 레지스터의 값을 대체합니다.

새로운  $x =$  이전  $x\{+, -, \times, \div\}$  변수

예를 들어, X 레지스터의 숫자(표시된 3)를 A(12)의 값으로 나눈다고 가정할 경우 RCL ÷ A를 누르면 이제  $x = 0.25$ 가 되지만 A에는 여전히 12가 들어 있습니다. 다시 호출 산술을 사용하면 프로그램의 메모리가 절약됩니다. RCL + A(1개 명령)에서 사용하는 메모리 양은 RCL A, +(2개 명령)의 절반입니다.



✓ 예:

변수 *D*, *E* 및 *F*에 값 1, 2, 3이 포함되어 있다고 가정합니다. 저장 산술을 사용하여 각 변수에 1을 더합니다.

키:	디스플레이:	설명:
<b>1</b> <b>STO</b> <b>D</b>	1.0000	가정된 값을 변수에 저장합니다.
<b>2</b> <b>STO</b> <b>E</b>	2.0000	
<b>3</b> <b>STO</b> <b>F</b>	3.0000	
<b>1</b> <b>STO</b>		<i>D</i> , <i>E</i> 및 <i>F</i> 에 1을 더합니다.
<b>+</b> <b>D</b> <b>STO</b>	1.0000	
<b>+</b> <b>E</b> <b>STO</b>		
<b>+</b> <b>F</b>		
<b>VIEW</b> <b>D</b>	D=	<i>D</i> 의 현재 값을 표시합니다.
	2.0000	
<b>VIEW</b> <b>E</b>	E=	
	3.0000	
<b>VIEW</b> <b>F</b>	F=	
	4.0000	
	1.0000	VIEW 디스플레이를 지우고 X 레지스터를 다시 표시합니다.

변수 *D*, *E* 및 *F*에 이전 예의 값 2, 3, 4가 포함되어 있다고 가정합니다. 3을 *D*로 나누고 그 결과에 *E*를 곱한 다음 *F*를 결과에 더합니다.

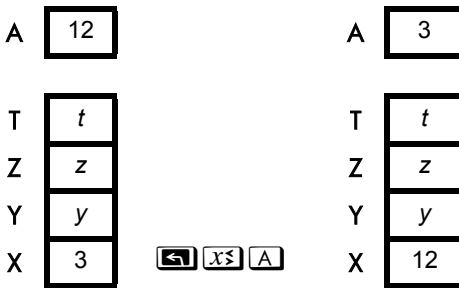
키:	디스플레이:	설명:
<b>3</b> <b>RCL</b> <b>÷</b> <b>D</b>	1.5000	3 ÷ <i>D</i> 를 계산합니다.
<b>RCL</b> <b>×</b> <b>E</b>	4.5000	3 ÷ <i>D</i> × <i>E</i> 입니다.
<b>RCL</b> <b>+</b> <b>F</b>	8.5000	3 ÷ <i>D</i> × <i>E</i> + <i>F</i> 입니다.

## x를 임의 변수와 교환

**x↔** 키를 사용하면 *x*의 내용(표시된 X 레지스터)을 임의 변수의 내용과 교환할 수 있습니다. 이 함수를 실행해도 Y, Z 또는 T 레지스터에는 영향을 주지 않습니다.

예:

키:	디스플레이:	설명:
<b>1</b> <b>2</b> <b>→</b> <b>STO</b>	12.0000	12를 변수 A에 저장합니다.
<b>A</b> <b>ENTER</b>		
<b>3</b>	3_	x를 표시합니다.
<b>←</b> <b>x↔</b> <b>A</b>	12.0000	X 레지스터의 내용과 변수 A를 교환합니다.
<b>←</b> <b>x↔</b> <b>A</b>	3.0000	X 레지스터의 내용과 변수 A를 교환합니다.



## 변수 "I" 및 "J"

직접 액세스할 수 있는 두 개의 변수(변수 I와 J)가 있습니다. 변수 I와 J는 다른 변수와 마찬가지로 값을 저장하지만 (I) 및 (J) 명령을 사용하여 통계 레지스터를 비롯한 다른 변수를 참조하는 데 사용할 수 있다는 점에서 특별합니다. (I)는 **0** 키에 있고 (J)는 **◦** 키에 있습니다. 이 프로그래밍 기술을 간접 주소 지정이라고 하며, 14장의 "변수 및 레이블의 간접 주소 지정"에서 설명합니다.





## 실수 함수

---

이 장에서는 프로그램에서 사용되는 일부 숫자 함수(예: 절대값 함수인 ABS)를 비롯하여 실수에 대해 계산을 수행하는 대부분의 계산기 함수를 다룹니다. 이러한 함수는 다음과 같이 그룹으로 설명됩니다.

- 지수 및 로그 함수
- 나눗셈의 몫과 나머지
- 거듭제곱 함수 ( $y^x$  및  $\frac{x}{y}$ )
- 삼각 함수
- 쌍곡선 함수
- 백분율 함수
- 물리상수
- 좌표, 각도 및 단위의 변환 함수
- 확률 함수
- 숫자의 특정 부분(숫자 변경 함수)

산술 함수와 계산에 대해서는 1장과 2장에서 다루었습니다. 고급 숫자 연산(근 찾기, 적분, 복소수, 진법 변환 및 통계)에 대해서는 이후 장에서 설명합니다. 이 장의 예에서는 모두 HP 35s가 RPN 모드에 있다고 가정합니다.

### ✓ 지수 및 로그 함수

숫자를 디스플레이에 표시한 다음 함수를 실행합니다. **[ENTER]**를 누를 필요가 없습니다.

계산:	키:
자연 로그( $e$ 진수)	LN
상용 로그(10진수)	LOG
자연 지수	$e^x$
상용 지수(역대수)	$10^x$

## ✓ 나눗셈의 몫과 나머지

**2** ( $2INT\div$ ) 및 **3** ( $3Rmdr$ )을 사용하여 두 정수의 나눗셈에서 각각 정수의 몫과 정수의 나머지를 구할 수 있습니다.

1. 첫 번째 정수를 입력합니다.
2. **ENTER**를 눌러 첫 번째 숫자와 두 번째 숫자를 구분합니다.
3. 두 번째 숫자를 입력합니다. **ENTER**를 누르지 *마십시오*.
4. 기능 키를 누릅니다.

예:

58 ÷ 9의 몫과 나머지를 표시합니다.

키:	디스플레이:	설명:
<b>ENTER</b>	6.0000	몫을 표시합니다.
<b>2</b> ( $2INT\div$ )		
<b>ENTER</b>	4.0000	나머지를 표시합니다.
<b>3</b> ( $3Rmdr$ )		

## ✓ 거듭제곱 함수

RPN 모드에서 숫자  $y$ 의  $x$ 제곱을 계산하려면  $y$  **ENTER**  $x$ 를 입력한 다음  $y^x$ 을 누릅니다.  $y > 0$ 이면  $x$ 는 임의의 숫자일 수 있고,  $y < 0$ 이면  $x$ 는 양수여야 합니다.

계산:	키:	결과:
152	<b>1</b> <b>5</b> <b>↵</b> <b>x<sup>2</sup></b>	225.0000
106	<b>6</b> <b>↵</b> <b>10<sup>x</sup></b>	1,000,000.0000
54	<b>5</b> <b>ENTER</b> <b>4</b> <b>y<sup>x</sup></b>	625.0000
2-1.4	<b>2</b> <b>ENTER</b> <b>1</b> <b>.</b> <b>4</b> <b>+/-</b> <b>y<sup>x</sup></b>	0.3789
(-1.4) <sup>3</sup>	<b>1</b> <b>.</b> <b>4</b> <b>+/-</b> <b>ENTER</b> <b>3</b> <b>y<sup>x</sup></b>	-2.7440

RPN 모드에서 숫자  $y$ 의  $x$ 제곱근( $y$ 의  $x$ 제곱근)을 계산하려면  $y$  **ENTER**  $x$ 를 누른 다음 **↵** **y<sup>x</sup>**를 누릅니다.  $y < 0$ 이면  $x$ 는 정수여야 합니다.

계산:	키:	결과:
$\sqrt{196}$	<b>1</b> <b>9</b> <b>6</b> <b>√x</b>	14.0000
$\sqrt[3]{-125}$	<b>1</b> <b>2</b> <b>5</b> <b>+/-</b> <b>ENTER</b> <b>3</b> <b>↵</b> <b>y<sup>x</sup></b>	-5.0000
$\sqrt[4]{625}$	<b>6</b> <b>2</b> <b>5</b> <b>ENTER</b> <b>4</b> <b>y<sup>x</sup></b>	5.0000
$-1.4\sqrt[3]{.37893}$	<b>.</b> <b>3</b> <b>7</b> <b>8</b> <b>9</b> <b>3</b> <b>ENTER</b> <b>1</b> <b>.</b> <b>4</b> <b>+/-</b> <b>↵</b> <b>y<sup>x</sup></b>	2.0000

## 삼각법

### $\pi$ 입력

**↵** **π**를 눌러  $\pi$ 의 처음 12자리수를 X 레지스터에 저장합니다.

표시되는 숫자는 디스플레이 형식에 따라 달라집니다. **↵** **π**는  $\pi$ 의 근사값을 스택에 반환하는 함수이므로 **ENTER**를 누를 필요가 없습니다.

$\pi$ 는 초월수이기 때문에 계산기에서 정확하게  $\pi$ 를 나타낼 수는 없습니다.

## 각도 모드 설정

각도 모드는 삼각 함수에 사용된 각도에 대해 가정할 측정 단위를 지정합니다. 이 모드에서 기존의 숫자는 변환되지 *않습니다*. 이 장의 뒷부분에 있는 “변환 함수”를 참조하십시오.

$$360\text{도} = 2\pi\text{라디안} = 400\text{그레드}$$

각도 모드를 설정하려면 **[MODE]**를 누릅니다. 옵션을 선택할 수 있는 메뉴가 표시됩니다.

옵션	설명	표시기
DEG	60분법 도(도, 분, 초) 대신 10진수 도를 사용하는 도 모드를 설정합니다.	없음
RAD	라디안 모드를 설정합니다.	<b>RAD</b>
GRAD	그라디언트 모드를 설정합니다.	<b>GRAD</b>

## ✓ 삼각 함수

디스플레이에  $x$ 가 표시된 상태에서 다음을 수행합니다.

계산:	키:
$x$ 의 사인	<b>[SIN]</b>
$x$ 의 코사인	<b>[COS]</b>
$x$ 의 탄젠트	<b>[TAN]</b>
$x$ 의 아크사인	<b>[↶] [ASIN]</b>
$x$ 의 아크코사인	<b>[↶] [ACOS]</b>
$x$ 의 아크탄젠트	<b>[↶] [ATAN]</b>

### 참고



계산기의 내부 전체 자릿수가 15자리이기 때문에 무리수  $\pi$ 를 사용한 계산은 계산기에서 *정확하게* 표현할 수 없습니다. 이러한 제한은 삼각법에서 특히 두드러집니다. 예를 들어, 계산된  $\sin \pi$ (라디안)는 0이 아니라 0에 가까운 매우 작은 숫자인  $-2.0676 \times 10^{-13}$ 입니다.

## 4-4 실수 함수

**예:**

코사인  $(5/7)\pi$  라디안과 코사인  $128.57^\circ$ 가 4개 유효 자릿수까지 같음을 표시합니다.

<b>키:</b>	<b>디스플레이:</b>	<b>설명:</b>
<b>MODE</b> <b>2</b> (2RAD)		라디안 모드를 설정합니다.
<b>0</b> <b>5</b> <b>0</b> <b>7</b> <b>ENTER</b>	0.7143	<b>RAD</b> 표시기가 켜집니다.
<b>←</b> <b>π</b> <b>x</b> <b>COS</b>	-0.6235	소수 형식의 $5/7$ 입니다.
<b>MODE</b> <b>1</b> (1DEG)	-0.6235	Cos $(5/7)\pi$ 입니다.
<b>1</b> <b>2</b> <b>8</b> <b>0</b> <b>5</b> <b>7</b>	-0.6235	도 모드로 전환합니다(표시 없음).
<b>COS</b>		$\cos (5/7)\pi$ 와 같은 $\cos 128.57^\circ$ 를 계산합니다.

**프로그래밍 참고:**

역삼각 함수를 사용하여 각도  $\theta$ 를 확인하는 방정식은 대체로 다음과 같이 나타냅니다.

$$\theta = \arctan (y/x)$$

$x = 0$ 이면  $y/x$ 가 정의되지 않으며 **DIVIDE BY 0**이라는 오류가 발생합니다.

# 쌍곡선 함수

디스플레이에  $x$ 가 표시된 상태에서 다음을 수행합니다.

계산:	키:
$x$ 의 쌍곡 사인(SINH)	HYP SIN
$x$ 의 쌍곡선 코사인(COSH)	HYP COS
$x$ 의 쌍곡선 탄젠트(TANH)	HYP TAN
$x$ 의 쌍곡선 아크사인(ASINH)	HYP  ASIN
$x$ 의 쌍곡선 아크코사인(ACOSH)	HYP  ACOS
$x$ 의 쌍곡선 아크탄젠트(ATANH)	HYP  ATAN

# ✓ 백분율 함수

백분율 함수는 백분율 계산의 결과를 X 레지스터에 반환할 때 진수의 값을 Y 레지스터에 유지하기 때문에 및 에 비해 특별합니다. 따라서 진수를 다시 입력하지 않고 진수와 결과를 모두 사용하여 이후 계산을 수행할 수 있습니다.

계산:	키:
$y$ 의 $x\%$	$y$ $x$ %
$y$ 에서 $x$ 로의 변동률( $y \neq 0$ )	$y$ $x$ %CHG

예:

15.76달러 항목의 총 비용과 6% 판매세를 구합니다.

비용이 적절하게 반올림되도록 FIX 2 디스플레이 형식을 사용합니다.

키:

디스플레이:

설명:

**←** **DISPLAY** **1**(1FIX)

**2**

**1** **5** **.** **7** **6** **ENTER**

**6** **→** **%**

**+**

15.76

0.95

16.71

디스플레이를 소수 두 자릿수까지 반올림합니다.

6% 세금을 계산합니다.

총 비용(기본 가격 + 6% 세금)입니다.

작년에 15.76달러 항목의 비용이 16.12달러였다고 가정합니다. 작년 가격에서 금년 가격으로의 변동률은 얼마입니까?

키:

디스플레이:

설명:

**1** **6** **.** **1** **2** **ENTER**

**1** **5** **.** **7** **6** **←**

**%CHG**

**←** **DISPLAY** **1**(1FIX)

**4**

16.12

-2.23

-2.2333

금년 가격이 작년 가격보다 약 2.2% 인하되었습니다.

FIX 4 형식으로 복원합니다.

---


### 참고



%CHG 함수에서는 두 숫자의 순서가 중요합니다. 순서는 변동률의 양수 또는 음수 지정에 영향을 줍니다.

---

# 물리상수

CONST 메뉴에는 41개의 물리상수가 있습니다.  **CONST** 를 눌러 다음 항목을 볼 수 있습니다.

## CONST 메뉴







항목	설명	값
c	진공에서의 빛의 속도	$299792458 \text{ m s}^{-1}$
g	표준중력가속도	$9.80665 \text{ m s}^{-2}$
G	뉴턴 중력상수	$6.673 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$
V <sub>m</sub>	이상기체의 몰부피	$0.022413996 \text{ m}^3 \text{ mol}^{-1}$
N <sub>A</sub>	아보가드로 상수	$6.02214199 \times 10^{23} \text{ mol}^{-1}$
R <sub>∞</sub>	리드버그 상수	$10973731.5685 \text{ m}^{-1}$
eV	기본전하	$1.602176462 \times 10^{-19} \text{ C}$
m <sub>e</sub>	전자 질량	$9.10938188 \times 10^{-31} \text{ kg}$
m <sub>p</sub>	양자 질량	$1.67262158 \times 10^{-27} \text{ kg}$
m <sub>n</sub>	중성자 질량	$1.67492716 \times 10^{-27} \text{ kg}$
m <sub>H</sub>	중성자 질량	$1.88353109 \times 10^{-28} \text{ kg}$
k	볼츠만 상수	$1.3806503 \times 10^{-23} \text{ J K}^{-1}$
h	플랑크 상수	$6.62606876 \times 10^{-34} \text{ J s}$
ħ	2 파이(pi)에 대한 플랑크 상수	$1.054571596 \times 10^{-34} \text{ J s}$
Φ <sub>0</sub>	자기다발양자	$2.067833636 \times 10^{-15} \text{ Wb}$
λ <sub>0</sub>	보어 반경	$5.291772083 \times 10^{-11} \text{ m}$
ε <sub>0</sub>	전기 상수	$8.854187817 \times 10^{-12} \text{ F m}^{-1}$
R	몰기체 상수	$8.314472 \text{ J mol}^{-1} \text{ K}^{-1}$
F	패러데이 상수	$96485.3415 \text{ C mol}^{-1}$
u	원자 질량 상수	$1.66053873 \times 10^{-27} \text{ kg}$
μ <sub>0</sub>	자기 상수	$1.2566370614 \times 10^{-6} \text{ NA}^{-2}$
μ <sub>B</sub>	보어 마그네톤	$9.27400899 \times 10^{-24} \text{ J T}^{-1}$
μ <sub>N</sub>	핵마그네톤	$5.05078317 \times 10^{-27} \text{ J T}^{-1}$
μ <sub>p</sub>	양자 자기 모멘트	$1.410606633 \times 10^{-26} \text{ J T}^{-1}$
μ <sub>e</sub>	전자 자기 모멘트	$-9.28476362 \times 10^{-24} \text{ J T}^{-1}$
μ <sub>n</sub>	중성자 자기 모멘트	$-9.662364 \times 10^{-27} \text{ J T}^{-1}$
μ <sub>H</sub>	중성자의 자기 모멘트	$-4.49044813 \times 10^{-26} \text{ J T}^{-1}$



항목	설명	값
$r_e$	고전전자반지름	$2.817940285 \times 10^{-15} \text{ m}$
$Z_0$	진공에서의 특성 임피던스	$376.730313461 \Omega$
$\lambda_C$	컴프턴 파장	$2.426310215 \times 10^{-12} \text{ m}$
$\lambda_{c\alpha}$	중성자의 컴프턴 파장	$1.319590898 \times 10^{-15} \text{ m}$
$\lambda_{c\beta}$	양자의 컴프턴 파장	$1.321409847 \times 10^{-15} \text{ m}$
$\alpha$	미세구조상수	$7.297352533 \times 10^{-3}$
$\sigma$	스테판-볼츠만 상수	$5.6704 \times 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$
$t$	섭씨 온도	273.15
$\text{atm}$	표준 기압	101325 Pa
$\mu_B$	양자 자기회전비	$267522212 \text{ s}^{-1} \text{ T}^{-1}$
$C_1$	제1 복사상수	$374177107 \times 10^{-16} \text{ W m}^2$
$C_2$	제2 복사상수	$0.014387752 \text{ m K}$
$G_0$	양자전도	$7.748091696 \times 10^{-5} \text{ S}$
$e$	자연 로그의 진수(자연 상수)	2.71828182846

참조: Peter J.Mohr and Barry N.Taylor, CODATA Recommended Values of the Fundamental Physical Constants: 1998, Journal of Physical and Chemical Reference Data, Vol.28, No.6, 1999 및 Reviews of Modern Physics, Vol.72, No.2, 2000.

### 상수를 삽입하려면

1. 상수를 삽입하려는 위치에 커서를 놓습니다.
2.  **[CONST]** 를 눌러 물리상수 메뉴를 표시합니다.
3.     를 누르거나  **[CONST]** 를 눌러 한 번에 한 페이지씩 다음 페이지에 액세스하여 원하는 상수에 밑줄이 그어질 때까지 메뉴를 스크롤한 다음 **[ENTER]** 를 눌러 상수를 삽입합니다.

수식, 방정식 및 프로그램에 사용하는 경우 값이 아닌 이름으로 상수를 참조해야 합니다.

# 변환 함수

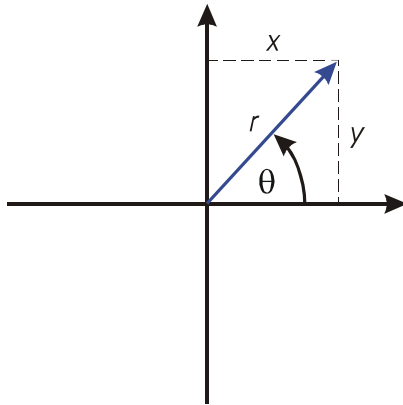
HP 35s에서는 4가지 변환 유형을 지원합니다. 다음을 서로 변환할 수 있습니다.

- 복소수의 직교좌표 및 극좌표 형식
- 각도 측정값의 도, 라디안 및 그라디언트
- 시간 및 도 각도의 10진수 및 60분법 형식
- 지원되는 다양한 단위(cm/인치, kg/파운드 등)

직교좌표 및 극좌표 변환을 제외하고 각 변환은 특정 키와 연결되어 있습니다. 키의 왼쪽(노란색) shift는 한 방향으로 변환하고, 동일한 키의 오른쪽(파란색) shift는 반대 방향으로 변환합니다. 이 유형의 각 변환에 대해 입력한 숫자는 다른 단위를 사용하여 측정된 것으로 가정됩니다. 예를 들어,  $\boxed{\rightarrow^{\circ}F}$ 를 사용하여 숫자를 화씨로 변환하는 경우 입력한 숫자는 섭씨로 측정된 온도라고 가정됩니다. 이 장의 예에서는 RPN 모드를 사용합니다. ALG 모드에서는 함수를 먼저 입력한 다음 변환할 숫자를 입력합니다.

## 직교좌표/극좌표 변환

극좌표( $r, \theta$ ) 및 직교좌표( $x, y$ )는 다음 그림과 같이 측정됩니다. 각도  $\theta$ 는 현재 각도 모드에 의해 설정된 단위를 사용합니다.  $\theta$ 에 대해 계산된 결과는  $-180^{\circ} - 180^{\circ}$ ,  $-\pi - \pi$  라디안 또는  $-200 - 200$  그라드 사이입니다.



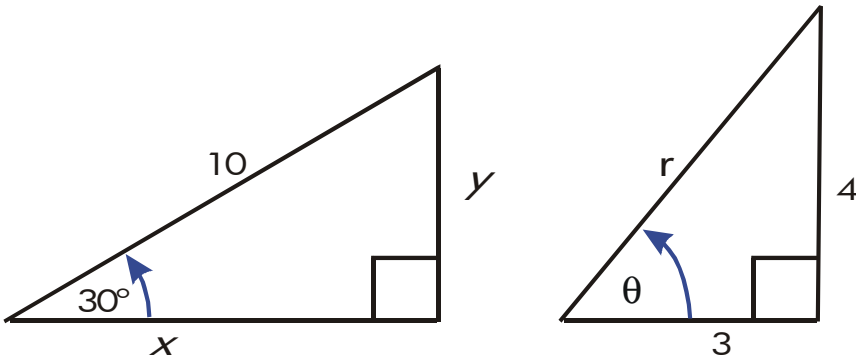
## 직교좌표와 극좌표를 변환하려면

복소수를 나타내는 형식은 모드 설정입니다. 임의 형식으로 복소수를 입력할 수 있으며, 입력 시 복소수는 모드 설정에 의해 결정된 형식으로 변환됩니다. 복소수 형식을 설정하는 데 필요한 단계는 다음과 같습니다.

1. RPN 모드에서  $\left[ \leftarrow \right]$   $\left[ \text{DISPLAY} \right]$  를 누르고  $\left[ 9 \right]$  ( $9 \times i \cdot y$ ) 또는  $\left[ \left[ \cdot \right] \left[ 0 \right] \right]$  ( $10r\theta a$ )를 선택합니다. ALG 모드에서는  $\left[ \left[ \cdot \right] \left[ 1 \right] \right]$  ( $11x+y \cdot i$ )를 선택할 수도 있습니다.
2. 원하는 좌표 값( $x$   $\left[ i \right]$   $y$ ,  $x$   $\left[ \sqrt{x} \right]$   $y$   $\left[ i \right]$  또는  $r$   $\left[ \left[ \cdot \right] \left[ \theta \right] \right]$   $a$ )을 입력합니다.
3.  $\left[ \text{ENTER} \right]$  를 누릅니다.

**예:** 극좌표를 직교좌표로 변환

다음 직각 삼각형에서 왼쪽에 있는 삼각형의 모서리  $x$ 와  $y$ 를 구하고 오른쪽 삼각형의 빗변  $r$ 과 각도  $\theta$ 를 구합니다.



**키:**

$\left[ \text{MODE} \right] \left[ 1 \right]$  (1DEG)  
 $\left[ \leftarrow \right]$   $\left[ \text{DISPLAY} \right] \left[ 9 \right]$  ( $9 \times i \cdot y$ )  
 $\left[ 1 \right] \left[ 0 \right] \left[ \left[ \cdot \right] \left[ \theta \right] \right] \left[ 3 \right] \left[ 0 \right]$   
 $\left[ \text{ENTER} \right]$

**디스플레이:**

8.660315.0000

**설명:**

도 및 복소수 좌표 모드를 설정합니다.

$r\theta a$ (극좌표)를  $xiy$ (직교좌표)로 변환합니다.

**DISPLAY** **0**

(10rθa)

**3** **i** **4** **ENTER**

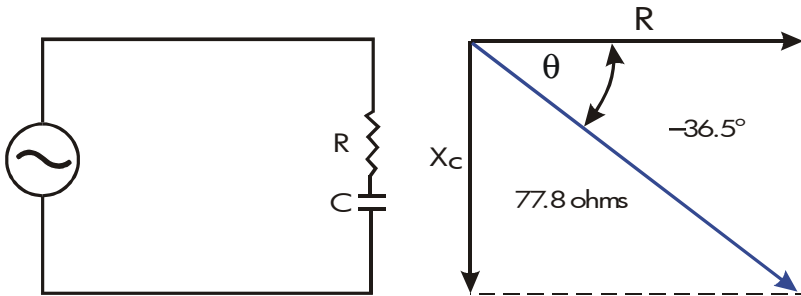
10.0000030.0000 복소수 좌표 모드를 설정합니다.

5.0000053.1301  $xy$ (직교좌표)를  $r\theta a$ (극좌표)로 변환합니다.

**예:** 벡터를 사용한 변환

엔지니어 P.C. Bord는 표시된 RC 회로에서 전체 임피던스가 77.8옴이고 전압이 전류보다 36.5°만큼 지연되는 것을 확인했습니다. 회로에서 저항 R과 용량 리액턴스  $X_C$ 의 값은 얼마입니까?

표시된 그림과 같이 임피던스가 극좌표 크기  $r$ 과 같고 전압 지연이 각도  $\theta$ (도)와 같은 벡터 다이어그램을 사용합니다. 값을 직교좌표로 변환하면  $x$ 값은  $R$ (옴)을 산출하고  $y$ 값은  $X_C$ (옴)을 산출합니다.



**키:**

**MODE** **1**(1DEG)

**PSE** **DISPLAY** **9**

(9x*i*,v)

**7** **7** **.** **8** **↵** **0** 77.80-36.5

**3** **6** **.** **5** **+/-**

**ENTER**

**디스플레이:**

62.5401*i*-46.2772

**설명:**

도 및 복소수 좌표 모드를 설정합니다.

전압 지연(도)인  $\theta$ 를 입력합니다. 전체 임피던스(옴)인  $r$ 를 입력합니다.

$x$ (옴)를 계산합니다

(저항  $R$ ).

$y$ (옴)를 계산합니다





(리액턴스  $X_C$ ).

**4-12 실수 함수**

## 시간 변환
















HP 35s는 숫자를 10진수 형식과 60분법 형식 간에 변환할 수 있습니다. 이 기능은 특히 도 단위로 측정된 각도와 시간에 유용합니다. 예를 들어, 10진수 형식에서 도 단위로 측정된 각도는 D.ddd...로 표현되는 반면, 60분법에서는 동일한 각도가 D.MMSSss로 표현됩니다. 여기서 D는 도 측정값의 정수부이고, ddd...는 도 측정값의 소수부이고, MM은 정수 분 수이고, SS는 초 수의 정수부이고, ss는 초 수의 소수부입니다.

### ✓ 10진수 형식과 시간/분/초 형식 사이에서 변환하려면

1. 변환하려는 숫자를 입력합니다.
2.   HMS를 눌러 시간/도, 분 및 초로 변환하거나   HMS를 눌러 다시 10진수 형식으로 변환합니다.

### 예: 시간 형식 변환

1시간의 1/7은 몇 분, 몇 초입니까? FIX 6 디스플레이 형식을 사용합니다.

키:	디스플레이:	설명:
  1 (1FIX)		
		
   	0.000000	FIX 6 디스플레이 형식을 설정합니다.
 	0 1/7	소수 형식의 1/7시간입니다.
  1 (1FIX)	0.000000	8분 34.29초와 같습니다.
	0.083429	
  1 (1FIX)	0.000000	FIX 4 형식으로 복원합니다.
	0.0834	

### ✓ 각도 변환

라디안으로 변환할 때는 X 레지스터의 숫자가 도라고 가정되고, 도로 변환할 때는 X 레지스터의 숫자가 라디안이라고 가정됩니다.

## 도 및 라디안 단위의 각도를 변환하려면

### 예

이 예에서는 각도 측정값 30°를  $\pi/6$ 라디안으로 변환합니다.

<p><b>키:</b></p> <p><b>3 0</b></p> <p><b>↵ PSE →RAD</b></p>	<p><b>디스플레이:</b></p> <p>0.0000</p> <p>30_</p> <p>0.0000</p> <p>0.5236</p>	<p><b>설명:</b></p> <p>각도를 도 단위로 입력합니다.</p> <p>라디안으로 변환합니다. <math>\pi/6</math>의 소수 근사값인 0.5236으로 결과를 읽습니다.</p>
---	---	--

## 단위 변환




HP 35s는 키보드에 →kg, →lb, →°C, →°F, →cm, →in, →l, →gal, →MILE, →KM 등 모두 10개의 단위 변환 함수가 있습니다.

변환할 값:	대상 단위:	키:	표시되는 결과:
1파운드	kg	<b>1</b> <b>↵</b> <b>→kg</b>	0.4536(kg)
1kg	파운드	<b>1</b> <b>↵</b> <b>→lb</b>	2.2046(파운드)
32°F	°C	<b>3 2</b> <b>↵</b> <b>→°C</b>	0.0000(°C)
100°C	°F	<b>1 0 0</b> <b>↵</b> <b>→°F</b>	212.0000(°F)
1인치	cm	<b>1</b> <b>↵</b> <b>→cm</b>	2.5400(cm)
100cm	인치	<b>1 0 0</b> <b>↵</b> <b>→in</b>	39.3701(인치)
1갤론	l	<b>1</b> <b>↵</b> <b>→l</b>	3.7854(리터)
1l	갤론	<b>1</b> <b>↵</b> <b>→gal</b>	0.2642(갤론)
1마일	KM	<b>1</b> <b>↵</b> <b>→KM</b>	1.6093(KM)
1KM	마일	<b>1</b> <b>↵</b> <b>→MILE</b>	0.6214(마일)



---

## 확률 함수

### ✓ 계승



표시된 음수가 아닌 정수  $x(0 \leq x \leq 253)$ 의 계승을 계산하려면   (오른쪽 shift  키)을 누릅니다.

### ✓ 감마



정수가 아닌  $x$ ,  $\Gamma(x)$ 의 감마 함수를 계산하려면  $(x - 1)$ 을 입력하고   을 누릅니다.  $x!$  함수는  $\Gamma(x + 1)$ 을 계산합니다.  $x$ 의 값은 음의 정수일 수 없습니다.

## 확률



### ✓ 조합

한 번에  $r$ 개 사용되는  $n$ 개 항목의 가능한 집합 수를 계산하려면  $n$ 을 먼저 입력하고   을 누른 다음  $r$ (음수가 아닌 정수만)을 입력합니다. 한 집합에서는 항목이 한 번만 발생하고 동일한  $r$ 개 항목의 다른 순서는 별도로 계산되지 않습니다.



### ✓ 순열

한 번에  $r$ 개 사용되는  $n$ 개 항목의 가능한 정렬 수를 계산하려면  $n$ 을 먼저 입력하고   을 누른 다음  $r$ (음수가 아닌 정수만)을 입력합니다. 한 정렬에서는 항목이 한 번만 발생하고 동일한  $r$ 개 항목의 다른 순서가 별도로 계산됩니다.

### ✓ 초기값

난수 생성기의 새 초기값으로 숫자를  $x$ 에 저장하려면   를 누릅니다.

### ✓ 난수 생성기

$0 < x < 1$  범위에서 난수를 생성하려면   를 누릅니다. 숫자는 균일하게 분포된 의사 난수 시퀀스의 일부입니다. D. Knuth, *The Art of Computer Programming*, vol. 2, *Seminumerical Algorithms*, London: Addison Wesley, 1981의 스펙트럼 테스트를 통과합니다.

RANDOM 함수는 초기값을 사용하여 난수를 생성합니다. 생성된 각 난수는 다음 난수의 초기값이 됩니다. 따라서 동일한 초기값에서 시작하여 난수 시퀀스를 반복할 수 있습니다. SEED 함수를 사용하여 새 초기값을 저장할 수 있습니다. 메모리를 지우면 초기값이 0으로 재설정됩니다. 초기값이 0이면 계산기에서 자체 초기값을 생성합니다.

**예: 직원 조합**

14명의 여자 직원과 10명의 남자 직원을 고용하고 있는 회사에서 6인으로 이루어진 안전 위원회를 구성하려고 합니다. 만들 수 있는 직원 조합 수는 몇 개입니까?

키:	디스플레이:	설명:
<b>2</b> <b>4</b> <b>ENTER</b> <b>6</b>	24	24명의 직원이 한 번에 6명씩
	6_	그룹화됩니다.
<b>↵</b> <b>nCr</b>	134,596.0000	가능한 전체 조합 수입니다.

무작위로 직원을 선택하는 경우 위원회에 6명의 직원이 포함될 확률은 얼마입니까? 이벤트의 확률을 구하려면 해당 이벤트에 대한 조합 수를 전체 조합 수로 나눕니다.

키:	디스플레이:	설명:
<b>1</b> <b>4</b> <b>ENTER</b> <b>6</b>	14	14명의 여자 직원이 한 번에 6명씩
	6_	그룹화됩니다.
<b>↵</b> <b>nCr</b>	3,003.0000	위원회에 6명의 여자 직원이 포함되는 조합 수입니다.
<b>x↔y</b>	134,596.0000	전체 조합 수를 X 레지스터로 다시 불러옵니다.
<b>÷</b>	0.0223	여자 직원의 조합 수를 전체 조합 수로 나누어 한 조합이 모두 여자 직원으로 구성될 확률을 구합니다.



**4-16 실수 함수**





## 숫자의 특정 부분

이러한 함수는 주로 프로그래밍에서 사용됩니다.


### 정수부

- ✓  $x$ 의 소수부를 제거하고 0으로 대체하려면  **INTG**  **6** (6IP)를 누릅니다. 예를 들어, 14.2300의 정수부는 14.0000입니다.

### 소수부


- ✓  $x$ 의 정수부를 제거하고 0으로 대체하려면  **INTG**  **5** (5FP)를 누릅니다. 예를 들어, 14.2300의 소수부는 0.2300입니다.

### 절대값

X 레지스터의 숫자를 절대값으로 대체하려면  **ABS**를 누릅니다. 복소수 및 벡터의 경우 다음과 같이 계산됩니다.



1.  $r\theta a$  형식의 복소수 절대값은  $r$ 입니다.
2.  $xiy$  형식의 복소수 절대값은  $\sqrt{x^2 + y^2}$ 입니다.
3. 벡터  $[A_1, A_2, A_3, \dots, A_n]$ 의 절대값은  $|A| = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$ 입니다.

### 인수 값

복소수의 인수를 추출하려면  **ARG**를 사용합니다. 복소수의 인수는 다음과 같습니다.

1.  $r\theta a$  형식에는  $a$ 입니다.
2.  $xiy$  형식에서는  $A \tan(y/x)$ 입니다.

### 부호 값

- ✓  $x$ 의 부호를 나타내려면  **INTG**  **1** (1SGM)을 누릅니다.  $x$  값이 음수이면 -1.0000이 표시되고, 0이면 0.0000이 표시되고, 양수이면 1.0000이 표시됩니다.

## 가장 큰 정수

✓ 주어진 숫자보다 작거나 같은 가장 큰 정수를 구하려면 **[←] [INTG] [4]** (4INTG)를 누릅니다.

### 예:

이 예에서는 숫자의 특정 부분을 추출하는 많은 연산을 요약해서 보여 줍니다.

계산:	키:	디스플레이:
2.47의 정수부	<b>[2] [.] [4] [7] [←] [INTG] [6]</b> (6IP)	2.0000
2.47의 소수부	<b>[2] [.] [4] [7] [←] [INTG] [5]</b> (5FP)	0.4700
-7의 절대값	<b>[7] [+/-] [↔] [ABS]</b>	7.0000
9의 부호 값	<b>[9] [←] [INTG] [1]</b> (1SGN)	1.0000
-5.3보다 작거나 같은 가장 큰 정수	<b>[5] [.] [3] [+/-] [←] [INTG] [4]</b> (4INTG)	-6.0000

RND 함수(**[↔] [RND]**)는 디스플레이 형식에 의해 지정된 자릿수로  $x$ 를 내부적으로 반올림합니다. 내부 숫자는 12자릿수로 표현됩니다. 분수 표시 모드의 RND 동작에 대한 자세한 내용은 5장을 참조하십시오.

## 분수

1장의 **분수 절**에서는 분수 입력, 표시 및 계산의 기초를 소개했습니다. 이 장에서는 이러한 항목에 대해 자세히 설명합니다. 분수 입력 및 표시에 대해 간략히 살펴보면 다음과 같습니다.

- 분수를 입력하려면  $\square \cdot \square$  두 번 누릅니다. 혼합된 숫자의 정수부 뒤에서 한 번 누르고, 숫자 소수부의 분자와 분모 사이에 다시 누릅니다.  $2\frac{3}{8}$ 을 입력하려면  $\square 2 \cdot \square 3 \cdot \square 8$ 을 누릅니다.  $5/8$ 를 입력하려면  $\square \cdot \square 5 \cdot \square 8$  또는  $\square 0 \cdot \square 5 \cdot \square 8$ 을 누릅니다.
- 분수 표시 모드를 켜고 끄려면  $\square \rightarrow$  **FDISP**를 누릅니다. 분수 표시 모드를 끄면 디스플레이가 디스플레이 메뉴를 통해 설정된 이전 표시 형식으로 돌아갑니다. 활성화된 경우 이 메뉴를 통해 다른 형식을 선택해도 분수 표시 모드가 꺼집니다.
- 함수는 이 장의 뒷부분에서 설명하는 RND를 제외하고 소수와 동일한 방식으로 분수 계산을 수행합니다.

이 장의 예에서는 달리 명시되지 않은 경우 모두 RPN 모드를 사용합니다.

## 분수 입력

분자가 분모보다 큰 가분수를 비롯하여 거의 모든 숫자를 키보드에서 분수로 입력할 수 있습니다.

예:

키:	디스플레이:	설명:
$\square \rightarrow$ <b>FDISP</b>		분수 표시 모드를 켭니다.
$\square 1 \cdot \square 5$ <b>ENTER</b>	1 1/2	1.5를 입력하면 분수로 표시됩니다.
$\square 1 \cdot \square 3 \cdot \square 4$ <b>ENTER</b>	1 3/4	1 3/4을 입력합니다.
$\square \rightarrow$ <b>FDISP</b>	1.7500	x를 소수로 표시합니다.
$\square \rightarrow$ <b>FDISP</b>	1 3/4	x를 분수로 표시합니다.

예제와 동일한 결과를 얻지 못했다면 분수 표시 형식을 실수로 변경했을 수 있습니다. 이 장의 뒷부분에 있는 “분수 표시 변경”을 참조하십시오.

다음 항목에는 올바른 입력 분수와 잘못된 입력 분수의 예가 더 포함되어 있습니다.

---

## 디스플레이의 분수

분수 표시 모드에서 숫자는 내부적으로 소수로 계산된 다음 허용되는 가장 정확한 분수를 사용하여 표시됩니다. 또한 정확도 표시기에 분수 부정확도의 방향이 해당 12자리 소수 값과 비교하여 표시됩니다. 대부분의 통계 레지스터는 예외이며, 항상 소수로 표시됩니다.

### 표시 규칙

표시되는 분수가 입력한 분수와 다를 수도 있습니다. 기본 상태에서 계산기는 소수를 다음 규칙에 따라 표시합니다. 규칙을 변경하려면 이 장의 뒷부분에 있는 “분수 표시 변경”을 참조하십시오.

- 숫자에는 정수부와 필요한 경우 분자가 분모보다 작은 진분수가 있습니다.
- 분모는 4095 이하입니다.
- 분수는 최대한 약분됩니다.

#### 예:

다음은 입력된 값과 표시되는 값의 예입니다. 비교를 위해 내부 12자리 소수 값도 표시되어 있습니다. 마지막 열의 ▲ 및 ▼ 표시기에 대해서는 아래에서 설명합니다.

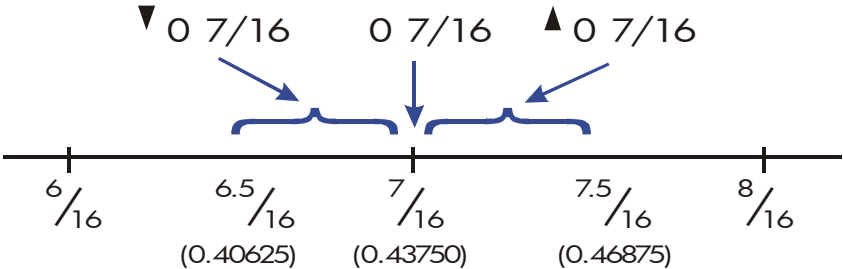
입력된 값	내부 값	표시되는 분수
$2\frac{3}{8}$	2.37500000000	$2\frac{3}{8}$
$14\frac{15}{32}$	14.4687500000	$14\frac{15}{32}$
$54/12$	4.50000000000	$4\frac{1}{2}$
$6\frac{18}{5}$	9.60000000000	$9\frac{3}{5}$
$34/12$	2.83333333333	$2\frac{5}{6}$ ▼
$15/8192$	0.00183105469	$0\frac{7}{3823}$ ▲
$12345678\frac{12345}{3}$	12349793.0000	12349793
$16\frac{3}{16384}$	16.0001831055	$16\frac{1}{4095}$

### 정확도 표시기

표시되는 분수의 정확도는 디스플레이 오른쪽에 있는 ▲ 및 ▼ 표시기로 나타납니다. 계산기는 내부 12자리 숫자의 소수부 값을 표시되는 분수 값과 비교합니다.

- 표시기가 켜지지 않는 경우 내부 12자리수 값의 소수부가 표시되는 분수 값과 정확하게 일치합니다.
- ▼가 켜지는 경우 내부 12자리수 값의 소수부가 표시되는 분수보다 약간 작습니다. 즉, *정확한 분자가 0.5 이내 범위에서 표시되는 분수보다 작습니다.*
- ▲가 켜지는 경우 내부 12자리수 값의 소수부가 표시되는 분수보다 약간 큼니다. 즉, *정확한 분자가 0.5 이내 범위에서 표시되는 분수보다 큼니다.*

이 다이어그램에서는 표시되는 분수가 근처 값과 어떻게 관련이 있는지를 보여줍니다. ▲는 정확한 분자가 표시되는 분자보다 “약간 위”에 있음을 나타내고, ▼는 정확한 분자가 “약간 아래”에 있음을 나타냅니다.



이 기능은 분수 표시 형식에 대한 규칙을 변경하는 경우에 특히 중요합니다. 뒷부분에 있는 “분수 표시 변경”을 참조하십시오. 예를 들어, 모든 분수에서 분모를 5로 적용하는 경우 정확한 분수가 대략  $3.3333/5$ 으로,  $3/5$ 보다 “약간 위”에 있으므로  $2/3$ 는  $\text{0 } 3/5 \blacktriangle$ 로 표시됩니다. 마찬가지로, 실제 분자가 3보다 “약간 위”에 있으므로  $-2/3$ 는  $-\text{0 } 3/5 \blacktriangle$ 로 표시됩니다.

때때로 표시기가 불필요하게 켜지는 경우도 있습니다. 예를 들어,  $2 \frac{2}{3}$ 를 입력하면 입력한 숫자와 정확하게 일치하지만  $2 \frac{2}{3} \blacktriangle$ 가 표시됩니다. 계산기는 항상 내부 값의 소수부를 분수의 12자릿수 값과 비교합니다. 내부 값에 정수부가 있는 경우 소수부의 자릿수가 12자릿수보다 작으며 12자릿수를 모두 사용하는 분수와 정확하게 일치할 수 없습니다.

---

## 분수 표시 변경

기본 상태에서 계산기는 소수를 특정 규칙에 따라 표시합니다. 그러나 원하는 분수 표시 형식에 따라 규칙을 변경할 수 있습니다.









- 사용되는 최대 분모를 설정할 수 있습니다.
- 세 가지 분수 형식 중 하나를 선택할 수 있습니다.

다음 몇 개 항목에서는 분수 표시를 변경하는 방법을 보여 줍니다.








### 최대 분모 설정

모든 분수에서 분모는 계산기에 저장된 값에 따라 선택됩니다. 분수를  $a \ b/c$ 로 간주할 때  $/c$ 가 분모를 제어하는 값입니다.

$/c$  값은 분수 표시 모드에서 사용되는 **최대** 분모만 정의합니다. 사용되는 특정 분모는 다음 항목에서 설명하는 분수 형식에 의해 결정됩니다.












- 최대 분모 값을 설정하려면 값을 입력하고  를 누릅니다. 분수 표시 모드가 자동으로 활성화됩니다. 4095를 초과하는 값은 입력할 수 없습니다.
- $\sqrt{x}$  값을 X 레지스터로 다시 호출하려면   를 누릅니다.
- 기본값을 4095로 복원하려면   를 누르거나 4095보다 큰 임의의 값을 최대 분모로 입력합니다. 다시 분수 표시 모드가 자동으로 활성화됩니다.

$\sqrt{x}$  함수는 X 레지스터에 있는 숫자 정수부의 절대값을 사용합니다. LAST X 레지스터의 값은 변경되지 않습니다.



표시되는 분수가 너무 길어 디스플레이에 들어가지 않는 경우  표시기가 나타납니다.   및  를 사용해서 페이지 단위로 스크롤하여 분수의 나머지 부분을 확인할 수 있습니다. 숫자의 소수 표현을 확인하려면 를 누른 다음 를 누르고 있습니다.


**예:**

이 예에서는 최대 분모를 3125로 설정한 다음 너무 길어 디스플레이에 들어가지 않는 분수를 표시하는 데 필요한 단계를 보여 줍니다.

키:	디스플레이:	설명:
    		최대 분모를 3125로 설정합니다.
		
   	0	분모에서 누락된 숫자를 확인합니다.
	1202604 888/31	니다.
	0	오른쪽으로 스크롤하여 분모의
	25	나머지 부분을 확인합니다.

**참고:**

1. ALG 모드에서는 줄 1에 수식을 입력한 다음  를 누를 수 있습니다. 이 경우 수식이 계산되고 그 결과를 사용하여 최대 분모가 결정됩니다.

2. ALG 모드에서는 계산 결과를 /c 함수의 인수로 사용할 수 있습니다. 값이 줄 2에 표시된 상태에서  /c 만 누르면 됩니다. 줄 2의 값이 분수 형식으로 표시되고 정수부를 사용하여 최대 분모가 결정됩니다.
3. 복소수나 벡터를 /c 명령의 인수로 사용할 수 없습니다. "INVALID DATA"라는 오류 메시지가 표시됩니다.

## 분수 형식 선택

계산기에는 세 가지 분수 형식이 있습니다. 표시되는 분수는 항상 선택된 형식에 대한 규칙 내에서 가장 정확한 분수입니다.

- **가장 정확한 분수.** 분수가 최대 /c 값까지 임의의 분모를 사용하며 최대한 약분됩니다. 예를 들어, 분수로 수학 개념을 공부하는 경우 모든 분모가 가능하도록 지정할 수 있습니다(/c 값은 4095임). 이것이 기본 분수 형식입니다.
- **분모의 인수.** 분수가 /c 값의 인수인 분모만 사용하며 최대한 약분됩니다. 예를 들어, 주가를 계산하는 경우  $53 \frac{1}{4}$  및  $37 \frac{7}{8}$ 를 표시할 수 있습니다(/c 값은 8임). 또는 /c 값이 12인 경우 가능한 분모는 2, 3, 4, 6 및 12입니다.
- **고정 분모.** 분수가 항상 /c 값을 분모로 사용하며 약분되지 않습니다. 예를 들어, 시간 측정값으로 작업하는 경우  $1 \frac{25}{60}$ 를 표시할 수 있습니다(/c 값은 60임).

분수 형식을 제어하는 3개의 플래그가 있습니다. 이러한 플래그에는 7, 8, 9라는 번호가 매겨져 있고 각 플래그는 해제되거나 설정됩니다. 플래그의 용도는 다음과 같습니다.



- 플래그 7은 분수 표시 모드를 켜거나 끕니다(해제 = 끄기, 설정 = 켜기).
- 플래그 8은 /c 값보다 작거나 같은 값 사용 또는 /c 값의 인수만 사용을 전환합니다(해제 = 아무 값이나 사용, 설정 = /c 값의 인수만 사용).
- 플래그 9는 플래그 8이 설정되어 있는 경우에만 작동하며, 분수 약분 여부를 전환합니다(해제 = 약분, 설정 = 약분 안 함).

플래그 8과 플래그 9를 적절하게 해제하거나 설정하면 아래 표와 같은 세 가지 분수 형식을 표시할 수 있습니다.










표시하려는 분수 형식:	다음 플래그 변경:	
	8	9
가장 정확한 형식	해제	—
분모의 인수	설정	해제
고정 분모	설정	설정

아래에 나열된 단계를 통해 플래그 8과 플래그 9를 변경하여 분수 형식을 설정할 수 있습니다. 플래그는 프로그램에서 특히 유용하기 때문에 14장에서 자세히 설명합니다.

1. **[ FLAGS]**를 눌러 플래그 메뉴를 표시합니다.
2. 플래그를 설정하려면 **[1](1SF)**를 누르고 플래그 번호(예: 8)를 입력합니다. 플래그를 해제하려면 **[2](2CF)**를 누르고 플래그 번호를 입력합니다. 플래그가 설정되었는지 확인하려면 **[3](3FS?)**를 누르고 플래그 번호를 입력합니다. YES 또는 NO 응답을 지우려면 **[C]** 또는 **[**를 누릅니다.

**예:**

이 예에서는 숫자  $\pi$ 를 사용하여 세 형식의 분수 표시를 보여 줍니다. 이 예에서는 분수 표시 형식이 활성화되었으며 플래그 8이 기본 상태(해제됨)에 있다고 가정합니다.

키:	디스플레이:	설명:
<b>[4][0][9][5][</b>		최대 /c 값을 다시 기본값으로 설정합니다.
<b>[/c]</b>		가장 정확한 형식입니다.
<b>[ <math>\pi</math>]</b>	0	플래그 8이 해제됩니다.
<b>[ FLAGS] [1](1SF)</b>	3 16/113	플래그 8이 설정됩니다.
<b>[8]</b>	0	분모의 인수 형식이며 819*5=4095입니다.
<b>[ FLAGS] [1](1SF)</b>	3 116/819	플래그 9가 설정됩니다.
<b>[9]</b>	0 0/4095	고정 분모 형식입니다.
<b>[ FLAGS] [2](2CF)</b>	3 580/4095	가장 정확한 기본 형식으로 돌아갑니다.
<b>[8][ FLAGS] [2](2CF)[9]</b>	0	
<b>[8][ FLAGS] [2](2CF)[9]</b>	3 16/113	

## 분수 표시 예

다음 표에서는 두 개의  $/c$  값에 대해 숫자 2.77이 세 분수 형식에서 어떻게 표시되는지를 보여 줍니다.

분수 형식	2.77의 표시 형식	
	$/c = 4095$	$/c = 16$
가장 정확한 형식	$2\ 77/100^{(2.7700)}$	$2\ 10/13\blacktriangle^{(2.7692)}$
분모의 인수	$2\ 1051/1365\blacktriangle^{(2.7699)}$	$2\ 3/4\blacktriangle^{(2.7500)}$
고정 분모	$2\ 3153/4095\blacktriangle^{(2.7699)}$	$2\ 12/16\blacktriangle^{(2.7500)}$

다음 표에서는  $/c$  값이 16인 경우 세 분수 형식에서 어떻게 다른 숫자가 표시되는지를 보여 줍니다.

분수 형식 *	입력된 숫자 및 표시되는 분수				
	2	2.5	$2\ 2/3$	2.9999	$216/25$
가장 정확한 형식	2	$2\ 1/2$	$2\ 2/3\blacktriangle$	$3\blacktriangledown$	$2\ 9/14\blacktriangledown$
분모의 인수	2	$2\ 1/2$	$2\ 11/16\blacktriangledown$	$3\blacktriangledown$	$2\ 5/8\blacktriangle$
고정 분모	$2\ 0/16$	$2\ 8/16$	$2\ 11/16\blacktriangledown$	$3\ 0/16\blacktriangledown$	$2\ 10/16\blacktriangle$

\*  $/c$  값이 16인 경우에 해당합니다.

## 분수 반올림

분수 표시 모드가 활성화 상태이면 RND 함수는 X 레지스터의 숫자를 분수의 가장 가까운 소수 표현으로 변환합니다. 반올림은 현재  $/c$  값과 플래그 8과 9의 상태에 따라 수행됩니다. 분수가 소수 표현과 정확하게 일치하면 정확도 표시기가 꺼집니다. 그렇지 않으면 정확도 표시기가 켜집니다. 이 장의 앞부분에 있는 “정확도 표시기”를 참조하십시오.

방정식이나 프로그램에서 분수 표시 모드가 활성화 상태이면 RND 함수는 소수 반올림을 수행합니다.

예:

56 3/4인치 공간이 있고 이 공간을 6개의 동일한 섹션으로 나누려고 합니다. 편리하게 1/16인치 증분으로 측정할 수 있다고 가정할 때 각 섹션의 너비는 얼마입니까? 누적 버림 오류는 얼마입니까?

키:	디스플레이:	설명:
<b>FLAGS</b> <b>8</b>		플래그 8을 설정합니다.
<b>1</b> <b>6</b>		1/16인치 증분에 대해 분수 형식을 설정합니다. 플래그 8과 플래그 9는 이전 예와 동일해야 합니다.
<b>5</b> <b>6</b> <b>.</b> <b>3</b> <b>.</b> <b>4</b>	56 3/4	거리를 D에 저장합니다.
<b>STO</b> <b>D</b>		
<b>6</b>	9 7/16▲	섹션이 9 7/16인치보다 약간 넓습니다.
<b>RND</b>	9 7/16	너비를 이 값으로 반올림합니다.
<b>6</b> <b>X</b>	56 5/8	6개 섹션의 너비입니다.
<b>RCL</b> <b>D</b> <b>-</b>	-0 1/8	누적 버림 오류입니다.
<b>FLAGS</b> <b>2</b> (2CF) <b>8</b>	-0 1/8	플래그 8을 해제합니다.
<b>FDISP</b>	-0.1250	분수 표시 모드를 끕니다.

## 방정식의 분수

방정식에 분수를 사용할 수 있습니다. 방정식을 표시할 때 방정식의 모든 숫자 값은 입력된 형태로 표시됩니다. 또한 방정식과 관련된 연산에 분수 표시 모드를 사용할 수 있습니다.

방정식을 계산 중이며 변수 값을 묻는 프롬프트가 표시되는 경우 분수를 입력할 수 있습니다. 값은 현재 표시 형식을 사용하여 표시됩니다.

방정식 작업에 대한 자세한 내용은 6장을 참조하십시오.

---

## 프로그램의 분수

방정식과 마찬가지로 프로그램에 분수를 사용할 수 있습니다. 숫자 값은 입력된 형태로 표시됩니다.

프로그램을 실행하는 경우 표시되는 값은 분수 표시 모드를 사용하여 표시됩니다 (활성화된 경우). INPUT 명령에서 값을 묻는 프롬프트를 표시하는 경우 분수를 입력할 수 있습니다. 프로그램의 결과는 현재 표시 형식을 사용하여 표시됩니다.

프로그램은  $/c$  함수를 사용하고 플래그 7, 8 및 9를 설정하고 해제하여 분수 표시를 제어할 수 있습니다. 14장의 “플래그”를 참조하십시오.

프로그램 작업에 대한 자세한 내용은 13장과 14장을 참조하십시오.

## 방정식 입력 및 계산

### 방정식을 사용할 수 있는 방법

다음과 같은 여러 방법으로 HP 35s에서 방정식을 사용할 수 있습니다.

- 계산할 방정식 지정(본 6장)
- 미지수 값을 구할 때 방정식 지정(7장)
- 적분할 함수 지정(8장)

#### 예: 방정식을 사용하여 계산

직선 파이프 섹션의 부피를 자주 확인해야 한다고 가정합니다. 방정식은 다음과 같습니다.

$$V = .25 \pi d^2 l$$

여기서  $d$ 는 파이프의 내부 지름이고  $l$ 은 파이프 길이입니다.

✓ 반복해서 계산을 입력할 수도 있습니다. 예를 들어,

$\cdot$   $2$   $5$   $\text{[ENTER]}$   $\leftarrow$   $\pi$   $\times$   $2$   $\cdot$   $5$   $\leftarrow$   $x^2$   $\times$   $1$   $6$   $\times$  는 지름이  $2\frac{1}{2}$  인치인 파이프 16인치의 부피를 계산합니다(78.5398입방 인치). 그러나 *방정식* 을 저장하면 여러 번 사용할 수 있도록 HP 35s에서 지름, 길이 및 부피의 관계를 “기억”합니다.

계산기를 방정식 모드로 설정하고 다음 키 입력을 사용하여 방정식을 입력합니다.

키:	디스플레이:	설명:
<b>EQN</b>	EQN LIST TOP	<b>EQN</b> 표시기로 나타나는 방정식 또는 줄 2의 현재 방정식
<b>RCL</b>		새 방정식을 시작합니다. <b>RCL</b> 은 변수 이름을 입력할 수 있도록 <b>A..Z</b> 표시기를 켭니다.
<b>V</b> <b>←</b> <b>=</b>	V=_	<b>RCL</b> <b>V</b> 는 <b>V</b> 를 입력합니다.
<b>.</b> <b>2</b> <b>5</b>	V=0.25_	숫자 입력 시 “_” 입력 커서가 사용됩니다.
<b>X</b> <b>←</b> <b>π</b> <b>X</b>	V=0.25×π×_	<b>X</b> 는 숫자를 끝냅니다.
<b>RCL</b> <b>D</b> <b>V<sup>x</sup></b> <b>2</b>	V=0.25×π×D^2_	<b>V<sup>x</sup></b> 는 ^를 입력합니다.
<b>X</b> <b>RCL</b> <b>L</b>	V=0.25×π×D^2×L_	
<b>ENTER</b>	V=0.25×π×D^2×L	방정식을 종료하고 표시합니다.
<b>←</b> <b>SHOW</b>	CK=49CA LN=14	키 입력을 확인할 수 있도록 방정식의 체크섬과 길이를 표시합니다.

방정식의 체크섬 및 길이를 예에 표시된 체크섬 및 길이와 비교하면 방정식을 올바르게 입력했는지 확인할 수 있습니다. 자세한 내용은 이 장의 끝에 있는 “방정식 확인”을 참조하십시오.

방정식 계산(V 계산):

키:	디스플레이:	설명:
<b>ENTER</b>	D? 값	방정식의 오른쪽에 변수를 묻는 프롬프트를 표시합니다. 먼저 <b>D</b> 를 묻는 프롬프트가 표시되며, 값은 <b>D</b> 의 현재 값입니다.
<b>2</b> <b>.</b> <b>1</b> <b>.</b> <b>2</b>	D? 2 1/2_	2 1/2인치를 분수로 입력합니다.
<b>R/S</b>	L? 값	<b>D</b> 를 저장하고 <b>L</b> 을 묻는 프롬프트를 표시합니다. 값은 <b>L</b> 의 현재 값입니다.
<b>1</b> <b>6</b> <b>R/S</b>	V= 78.5398	<b>L</b> 을 저장합니다. <b>V</b> (입방 인치)를 계산하고 그 결과를 <b>V</b> 에 저장합니다.






















## 6-2 방정식 입력 및 계산

## 방정식 연산 요약

만들어진 모든 방정식은 **방정식 목록에 저장됩니다**. 이 목록은 방정식 모드를 활성화할 때마다 표시됩니다.

특정 키를 사용하여 방정식과 관련된 연산을 수행합니다. 이러한 키에 대해서는 나중에 자세히 설명합니다.

방정식 목록에 방정식을 표시하는 경우 한 번에 두 개의 방정식이 표시됩니다. 현재 활성 방정식은 줄 2에 표시됩니다.

키	연산
	방정식 모드를 시작하고 종료합니다.
	표시된 방정식을 계산합니다. 방정식이 <i>지정</i> 인 경우 우변을 계산하고 그 결과를 좌변의 변수에 저장합니다. 방정식이 <i>등식</i> 이나 <i>수식</i> 인 경우 <b>XEQ</b> 와 같이 해당 값을 계산합니다. 이 장의 뒷부분에 있는 “방정식 유형”을 참조하십시오.
	표시된 방정식을 계산합니다. “=”가 있는 경우 “=”를 “_”로 대체하여 해당 값을 계산합니다.
	지정한 미지수 변수에 대해 표시된 방정식을 푼다 (7장 참조).
	지정한 변수와 관련해서 표시된 방정식을 적분합니다 (8장 참조).
	현재 방정식을 삭제하거나 커서 왼쪽에 있는 원소를 삭제합니다.
 또는 	표시된 방정식의 편집을 시작하며, 커서만 이동하고 내용은 삭제하지 않습니다.
  또는 	현재 방정식 표시 화면을 스크롤합니다.
 또는 	방정식 목록에서 단계별로 위나 아래로 이동합니다.
  또는 	방정식 목록의 맨 위나 맨 아래로 이동합니다.
 	
 <b>SHOW</b>	표시된 방정식의 체크섬(확인 값)과 길이(메모리 바이트)를 표시합니다.
 <b>UNDO</b>	가장 최근에 삭제된 원소나 방정식을 복구합니다.
	방정식 모드를 종료합니다.

프로그램에 방정식을 사용할 수도 있습니다. 이 내용에 대해서는 13장에서 설명합니다.

---

## 방정식 목록에 방정식 입력

방정식 목록은 입력한 방정식 모음입니다. 이 목록은 계산기의 메모리에 저장됩니다. 입력한 각 방정식은 자동으로 방정식 목록에 저장됩니다.

### 방정식을 입력하려면

원하는 대로 방정식 길이를 지정할 수 있으며, 사용 가능한 메모리 양에 의해서만 제한됩니다.

1. 계산기가 정상 작동 모드로 되어 있는지 확인합니다. 대체로 디스플레이에 숫자가 표시되면 정상입니다. 예를 들어, 변수 또는 프로그램 카탈로그가 표시되면 안 됩니다.
2. **[EQN]**을 누릅니다. **EQN** 표시기는 방정식 모드가 활성화 상태임을 나타내며, 방정식 목록의 항목이 표시됩니다.
3. 방정식 입력을 시작합니다. 이전 디스플레이가 입력하는 방정식으로 대체됩니다. 이전 방정식은 영향을 받지 않습니다. 실수를 한 경우 필요에 따라 **[←]** 또는 **[←] [UNDO]**를 누릅니다.
4. **[ENTER]**를 눌러 방정식을 종료하고 디스플레이에 표시합니다. 방정식이 자동으로 방정식 목록에 저장되며, 입력을 시작할 때 표시된 항목 바로 뒤에 저장됩니다. **[C]**를 대신 누르면 방정식이 저장되지만 방정식 모드가 꺼집니다.

방정식에는 변수, 숫자, 벡터, 함수 및 괄호가 포함될 수 있으며, 이 내용에 대해서는 다음 항목에서 설명합니다. 뒤에 나오는 예에서는 이러한 원소를 보여 줍니다.

### 방정식의 변수





방정식에 계산기의 모든 변수(**A-Z**, **(I)** 및 **(J)**)를 사용할 수 있습니다. 각 변수를 원하는 횟수만큼 사용할 수 있습니다. **(I)** 및 **(J)**에 대한 자세한 내용은 14장의 “변수 및 레이블에 간접 주소 지정”을 참조하십시오.

방정식에 변수를 입력하려면 **[RCL]** 변수를 누릅니다. **[RCL]**을 누르면 **A..Z** 표시기가 나타나 변수 키를 눌러 해당 이름을 방정식에 입력할 수 있음을 표시합니다.



## 방정식의 숫자



2, 8 및 16진수, 실수, 복소수 및 소수를 비롯한 모든 유효한 숫자를 방정식에 입력할 수 있습니다. 숫자는 항상 최대 12자를 표시하는 ALL 디스플레이 형식을 사용하여 표시됩니다.

방정식에 숫자를 입력하려면 ,  및 를 비롯한 표준 숫자 입력 키를 사용할 수 있습니다. 빼기 연산에 를 사용하지 마십시오.

## 방정식의 함수

방정식에 HP 35s 함수를 입력할 수 있습니다. 전체 목록은 이 장의 뒷부분에 있는 “방정식 함수”에 나와 있습니다. 부록 G, “연산 색인”에서도 이 정보를 제공합니다.

방정식을 입력할 때 일반적인 대수 방정식에 함수를 입력할 때와 거의 동일한 방식으로 함수를 입력합니다.

- 방정식에서 특정 함수는 대개 해당 인수 *사이에 표시됩니다*(예: “+” 및 “+”). 이러한 *사이* 연산자의 경우 동일한 순서로 방정식에 입력합니다.
- 다른 함수는 대개 하나 이상의 인수가 함수 이름 *뒤에* 있습니다(예: “COS” 및 “LN”). 이러한 *점두부* 함수의 경우 함수가 발생하는 방정식의 위치에 함수를 입력합니다. 키를 누르면 인수를 입력할 수 있도록 함수 이름 뒤에 왼쪽 괄호가 표시됩니다.
- 함수에 둘 이상의 인수가 있는 경우  을 눌러 구분합니다.

## 방정식의 괄호

방정식에 괄호를 포함하여 연산이 수행되는 순서를 제어할 수 있습니다. 괄호를 삽입하려면 **( )**를 누릅니다. 자세한 내용은 이 장의 뒷부분에 있는 “연산자 우선 순위”를 참조하십시오.

### 예: 방정식 입력

방정식  $r = 2 \times c \times (t - a) + 25$ 를 입력합니다.

키:	디스플레이:	설명:
<b>EQN</b>	$V=0.25 \times \pi \times D^2 \times L$	마지막으로 사용된 방정식을 방정식 목록에 표시합니다.
<b>RCL</b> <b>R</b> <b>↵</b> <b>=</b>	R=_	변수 R을 사용하여 새로운 방정식을 시작합니다.
<b>2</b>	R= 2_	숫자를 입력합니다.
<b>×</b> <b>RCL</b> <b>C</b> <b>×</b>	R=2×C×_	사이 연산자를 입력합니다.
<b>( )</b>	R=2×C×( )	왼쪽 괄호와 함께 접두부 함수를 입력합니다.
<b>RCL</b> <b>T</b> <b>-</b> <b>RCL</b>		인수 및 오른쪽 괄호를 입력합니다.
<b>A</b> <b>&gt;</b> <b>+</b> <b>2</b> <b>5</b>	=2×C×(T-A)+ 25_	
<b>ENTER</b>	R=2×C×(T-A)+25	방정식을 종료하고 표시합니다.
<b>↵</b> <b>SHOW</b>	CK=9E5F LN=14	체크섬과 길이를 표시합니다.
<b>C</b>		방정식 모드를 종료합니다.

---

## 방정식 표시 및 선택

방정식 목록에는 두 개의 기본 제공 방정식인 2\*2 lin. solve 및 3\*3 lin. Solve와 입력한 방정식이 들어 있습니다. 방정식을 표시하고 사용할 방정식을 선택할 수 있습니다.

### 6-6 방정식 입력 및 계산

## 방정식을 표시하려면

1. **[EQN]**을 누릅니다. 방정식 모드가 활성화되고 **EQN** 표시기가 켜집니다. 디스플레이에 방정식 목록의 항목이 표시됩니다.
  - **EQN LIST TOP**(방정식 포인터가 목록의 맨 위에 있는 경우)
  - 현재 방정식(마지막으로 표시된 방정식)
2. **[^]** 또는 **[v]**를 눌러 방정식 목록을 단계별로 이동하고 각 방정식을 표시합니다. 목록의 맨 위와 맨 아래가 “래핑”됩니다. **EQN LIST TOP**은 목록의 “맨 위”를 표시합니다.

## 긴 방정식을 보려면

1. 위에서 설명한 대로 방정식 목록에 방정식을 표시합니다. 14자보다 길면 14자만 표시됩니다. ➡ 표시기는 오른쪽에 숫자가 더 있음을 나타냅니다.
2. **[>]**를 눌러 방정식의 시작 부분에서 편집을 시작하거나 **[<]**를 눌러 방정식의 끝에서 편집을 시작합니다. 그런 다음 **[<]** 또는 **[>]**를 반복해서 눌러 방정식에서 커서를 한 번에 한 자씩 이동합니다. 왼쪽 또는 오른쪽에 문자가 더 있으면 **←** 및 **→**가 표시됩니다.
3. **[↔]<** 또는 **[↔]>**를 눌러 줄 2의 긴 방정식을 한 화면씩 스크롤합니다.

## 방정식을 선택하려면

위에서 설명한 대로 방정식 목록에 방정식을 표시합니다. 줄 2에 표시된 방정식은 모든 방정식 연산에 사용되는 방정식입니다.

## 예: 방정식 보기

마지막으로 입력한 방정식을 표시합니다.

키:	디스플레이:	설명:
<b>[EQN]</b>	$R=2 \times C \times (T-A)+25$	방정식 목록에 현재 방정식을 표시합니다.
<b>[&gt;]</b>	$R=2 \times C \times (T-A)+25$	방정식 왼쪽에 커서를 활성화합니다.
<b>[ENTER] &lt;</b>	$=2 \times C \times (T-A)+25_$	방정식 오른쪽에 커서를 활성화합니다.
<b>[C]</b>		방정식 모드를 종료합니다.

## 방정식 편집 및 지우기

입력 중인 방정식을 편집하거나 지울 수 있습니다. 방정식 목록에 저장된 방정식을 편집하거나 지울 수도 있습니다. 그러나 두 개의 기본 제공 방정식인  $2 \times 2$  lin. solve 및  $3 \times 3$  lin. solve는 편집하거나 지울 수 없습니다. 두 기본 제공 방정식 사이에 방정식을 삽입하려고 하면 새 방정식이  $3 \times 3$  lin. solve 뒤에 삽입됩니다.

### 입력 중인 방정식을 편집하려면

1.  $\leftarrow$  또는  $\rightarrow$ 를 눌러 커서 앞에 문자를 삽입할 수 있도록 커서를 이동합니다.
2. 커서를 이동하고  $\leftarrow$ 를 반복해서 눌러 불필요한 숫자나 함수를 삭제합니다. 방정식 편집 줄이 비어 있을 때는  $\leftarrow$ 를 눌러도 아무 영향을 주지 않지만, 빈 방정식 줄에서  $\text{ENTER}$ 를 누르면 빈 방정식 줄이 삭제됩니다. 그런 다음 디스플레이에 방정식 목록의 이전 항목이 표시됩니다.
3.  $\text{ENTER}$ (또는  $\text{C}$ )를 눌러 방정식을 방정식 목록에 저장합니다.

### 저장된 방정식을 편집하려면


1. 원하는 방정식을 표시하고,  $\rightarrow$ 를 눌러 방정식 시작 부분에 커서를 활성화하거나  $\leftarrow$ 를 눌러 방정식 끝에 커서를 활성화합니다(위의 “방정식 표시 및 선택” 참조).
2. 방정식에서 커서가 활성화 상태이면 새 방정식을 입력할 때와 마찬가지로 방정식을 편집할 수 있습니다.
3.  $\text{ENTER}$ (또는  $\text{C}$ )를 눌러 편집한 방정식을 방정식 목록에 저장하고 이전 버전을 대체합니다.

### 방정식을 편집하는 동안 메뉴 사용



1. 방정식을 편집할 때 설정 메뉴(예:  $\text{MODE}$ ,  $\leftarrow$   $\text{DISPLAY}$  또는  $\rightarrow$   $\text{CLEAR}$ )를 선택하면 방정식 편집 상태가 끝납니다.
2. 방정식을 편집할 때 삽입 또는 보기 메뉴(예:  $\text{L.R.}$ ,  $\leftarrow$   $\text{x,y}$ ,  $\rightarrow$   $\text{S.}\sigma$ ,  $\rightarrow$   $\text{SUMS}$ ,  $\rightarrow$   $\text{BASE}$ ,  $\leftarrow$   $\text{LOGIC}$ ,  $\text{RI}$ ,  $\leftarrow$   $\text{MEM}$  및  $\leftarrow$   $\text{CONST}$ )를 선택하면 항목을 삽입한 후에도 방정식이 편집 모드에 있습니다.
3. 메뉴  $\text{x?y}$ ,  $\text{FLAGS}$ ,  $\rightarrow$   $\text{x?0}$ 은 방정식 모드에서 사용할 수 없습니다.

## 6-8 방정식 입력 및 계산

## 저장된 방정식을 지우려면








원하는 방정식이 디스플레이의 줄 2에 있을 때까지 방정식 목록을 위나 아래로 스크롤한 다음 를 누릅니다.

## 저장된 모든 방정식을 지우려면

EQN 모드에서  **CLEAR**를 누릅니다. **3** (**3EQN**)을 선택합니다. **CLR EQN? Y**  
**N** 메뉴가 표시됩니다.  (Y) **ENTER**를 선택합니다.

## 예: 방정식 편집

이전 예에서 방정식의 25를 제거합니다.

키:	디스플레이:	설명:
	$R=2 \times C \times (T-A)+25$	방정식 목록에 현재 방정식을 표시합니다.
	$=2 \times C \times (T-A)+25\_$	방정식의 끝에 커서를 활성화합니다.
  	$=2 \times C \times \text{COS}(T-A)\_$	숫자 25를 삭제합니다.
	$R=2 \times C \times (T-A)$	방정식 목록에 편집된 방정식의 끝을 표시합니다.
		방정식 모드를 종료합니다.

---

## 방정식 유형

HP 35s는 3가지 방정식 유형에서 작동합니다.

- **등식.** 방정식에 "="가 포함되며 좌변에 단일 변수 이상이 포함됩니다. 예를 들어,  $x^2 + y^2 = r^2$ 은 등식입니다.
- **지정.** 방정식에 "="가 포함되며 좌변에 단일 변수만 포함됩니다. 예를 들어,  $A = 0.5 \times b \times h$ 는 지정입니다.
- **수식.** 방정식에 "="가 포함되지 *않습니다*. 예를 들어,  $x^3 + 1$ 은 수식입니다.

방정식을 *사용하여* 계산하는 경우 유형에 따라 계산 방법이 달라질 수는 있지만 모든 유형의 방정식을 사용할 수 있습니다. 미지수 변수에 대한 문제를 풀 때 등식 또는 지정을 사용합니다. 함수를 적분하는 경우 수식을 사용합니다.

---

## 방정식 계산

가장 유용한 방정식의 특징 중 하나는 *계산*을 통해 숫자 값을 생성할 수 있다는 점입니다. 이 기능을 통해 방정식의 결과를 계산할 수 있습니다. 또한 7장과 8장에서 설명하는 대로 방정식을 풀고 적분할 수 있습니다.

대부분의 방정식은 두 개의 변수가 "="로 구분되어 있으므로 방정식의 기본값은 두 변수의 값의 *차*입니다. 이 계산에서 방정식의 "="는 기본적으로 "-"로 처리됩니다. 값은 방정식의 균형이 얼마나 잘 조정되는지의 측정값입니다.

HP 35s에는 방정식 계산을 위한 두 개의 키인 **ENTER**와 **XEQ**가 있습니다. 두 키의 동작은 *지정* 방정식을 계산하는 방법에서만 차이가 있습니다.

- **XEQ**는 방정식 유형에 관계없이 방정식의 값을 반환합니다.
- **ENTER**는 *지정* 유형 방정식이 *아닌* 경우 방정식의 값을 반환합니다. 지정 방정식에서 **ENTER**는 우변의 값만 반환하고 해당 값을 좌변의 변수에 "입력"합니다. 즉, 값을 변수에 저장합니다.

다음 표에서는 방정식을 계산하는 두 가지 방법을 보여 줍니다.

방정식 유형	<b>ENTER</b> 의 결과	<b>XEQ</b> 의 결과
등식: $g(x) = f(x)$ 예: $x^2 + y^2 = r^2$	$g(x) - f(x)$ $x^2 + y^2 - r^2$	
지정: $y = f(x)$ 예: $A = 0.5 \times b \times h$	$f(x)^*$ $0.5 \times b \times h^*$	$y - f(x)$ $A - 0.5 \times b \times h$
수식: $f(x)$ 예: $x^3 + 1$	$f(x)$ $x^3 + 1$	
* 또한 결과를 좌변의 변수(예: A)에 저장합니다.		

### 방정식을 계산하려면

- 원하는 방정식을 표시합니다. 위의 “방정식 표시 및 선택”을 참조하십시오.
- ENTER** 또는 **XEQ**를 누릅니다. 방정식에서 필요한 각 변수의 값을 묻는 프롬프트를 표시합니다. 방정식의 진수가 현재 진수와 다른 경우 계산기에서 자동으로 결과를 현재 진수로 변경합니다.
- 각 프롬프트에 대해 원하는 값을 입력합니다.
  - 표시되는 값이 양호한 경우 **R/S**를 누릅니다.
  - 다른 값을 원하는 경우 값을 입력하고 **R/S**를 누릅니다. 이 장의 뒷부분에 있는 “방정식 프롬프트에 응답”을 참조하십시오.

계산을 중지하려면 **C** 또는 **R/S**를 누릅니다. **INTERRUPTED**라는 메시지가 줄 2에 표시됩니다.

방정식 계산은 스택의 값을 사용하지 않으며 방정식의 숫자와 변수 값만 사용합니다. 방정식의 값은 X 레지스터에 반환됩니다.

### 계산을 위해 ENTER 사용

방정식 목록에 방정식이 표시되면 **ENTER**를 눌러 방정식을 계산할 수 있습니다. 방정식을 입력 중이면 **ENTER**를 눌러도 방정식이 끝나기만 하고 계산되지 않습니다.

- 방정식이 지정인 경우 우변만 계산됩니다. 결과가 X 레지스터에 반환되고 좌변 변수에 저장된 다음 변수가 디스플레이에 표시됩니다. 궁극적으로 **ENTER** 는 좌변 변수의 값을 구합니다.
- 방정식이 등식이나 수식이면 **XEQ** 의 경우와 마찬가지로 전체 방정식이 계산됩니다. 결과는 X 레지스터에 반환됩니다.

### 예: ENTER를 사용하여 방정식 계산

이 장의 앞부분에 있는 방정식을 사용하여 지름이 35mm인 파이프 20미터 길이의 부피를 구합니다.

키:	디스플레이:	설명:
<b>EQN</b> (필요한 경우 <b>^</b> ) <b>ENTER</b>	$V=0.25 \times \pi \times D^2 \times L$  D? 2.5	원하는 방정식을 표시합니다.  값이 <b>W</b> 에 저장되도록 지정 방정식의 계산을 시작합니다. 방정식의 오른쪽에 변수를 묻는 프롬프트를 표시합니다. <b>D</b> 의 현재 값은 2.5입니다.
<b>3</b> <b>5</b> <b>R/S</b>	L? 16	<b>D</b> 를 저장하고 <b>L</b> 을 묻는 프롬프트를 표시합니다. <b>L</b> 의 현재 값은 16입니다.
<b>2</b> <b>0</b> <b>x</b> <b>1</b> <b>0</b> <b>0</b> <b>0</b> <b>ENTER</b> <b>R/S</b>	V= 19,242.255,0033	<b>L</b> (밀리미터)을 저장합니다. <b>V</b> (입방 밀리미터)를 계산하고 그 결과를 <b>W</b> 에 저장하며 <b>V</b> 를 표시합니다.
<b>÷</b> <b>1</b> <b>E</b> <b>6</b> <b>ENTER</b>	19.2423	입방 밀리미터를 리터로 변경하지만 <b>V</b> 를 변경하지 않습니다.

## 계산을 위해 XEQ 사용

방정식 목록에 방정식이 표시되면 **XEQ**를 눌러 방정식을 계산할 수 있습니다. 방정식 유형에 관계없이 전체 방정식이 계산됩니다. 결과는 X 레지스터에 반환됩니다.

## 6-12 방정식 입력 및 계산



## 예: XEQ를 사용하여 방정식 계산

이전 예의 결과를 사용하여 지름이 35.5mm로 변경되면 파이프 부피가 얼마나 변하는지 확인합니다.

키:	디스플레이:	설명:
<b>[EQN]</b>	$V=0.25 \times P \times D^2 \times L$	원하는 방정식을 표시합니다.
<b>[XEQ]</b>	V? 19,242,255.0033	방정식 계산을 시작하여 값을 구합니다. 모든 변수를 묻는 프롬프트를 표시합니다.
<b>[R/S]</b>	D? 35	동일한 V를 유지하고 D를 묻는 프롬프트를 표시합니다.
<b>[3] [5] [.] [5]</b>	L? 20,000	새로운 D를 저장하고 l을 묻는 프롬프트를 표시합니다.
<b>[R/S]</b>	-553,705.7051	동일한 l을 유지하고 방정식의 값을 계산합니다. 좌변과 우변의 값이 다릅니다.
<b>[R/S]</b>	-0.5537	입방 밀리미터를 리터로 변경합니다.
<b>[÷] [1] [E] [6]</b> <b>[ENTER]</b>		

방정식의 값은 이전 부피(V)에서 새로운 부피(새로운 D 값을 사용하여 계산됨)를 빼 값이므로 이전 부피가 표시된 양만큼 더 작습니다.

## 방정식 프롬프트에 응답

방정식에서 필요한 각 변수의 값을 묻는 프롬프트를 표시합니다. 프롬프트는 변수 이름과 현재 값을 제공합니다(예: X?2.5000). 명명되지 않은 간접 변수 (I) 또는 (J)가 방정식에 있는 경우 명명되지 않은 간접 변수에 저장된 현재 값이 자동으로 사용되므로 해당 값을 묻는 프롬프트는 표시되지 않습니다(14장 참조).

- 숫자를 변경하지 않고 그대로 두려면 **[R/S]**를 누릅니다.

- **숫자를 변경하려면** 새 숫자를 입력하고 **[R/S]**를 누릅니다. 새로운 숫자가 X 레지스터의 기존 값을 덮어씁니다. 필요한 경우 숫자를 분수로 입력할 수 있습니다. 숫자를 계산해야 하는 경우 일반적인 키보드 계산을 사용한 다음 **[R/S]**를 누릅니다. 예를 들어, RPN 모드에서 2 **[ENTER]** 5 **[y<sup>x</sup>]** **[R/S]**를 누르거나 ALG 모드에서 2 **[y<sup>x</sup>]** 5 **[ENTER]** **[R/S]**를 누를 수 있습니다. **[ENTER]**를 누르기 전에는 수식이 줄 2에 표시되고, **[ENTER]**를 누른 후에는 수식의 결과가 줄 2에 표시됩니다.
- **프롬프트를 취소하려면** **[C]**를 누릅니다. 변수의 현재 값은 X 레지스터에 유지되고 줄 2의 우변에 표시됩니다. 숫자 입력 중에 **[C]**를 누르면 숫자가 0으로 지워집니다. 프롬프트를 취소하려면 **[C]**를 다시 누릅니다.
- **프롬프트에 의해 숨겨진 숫자를 표시하려면** **[<img alt="show icon" data-bbox="578 311 615 328"/>]** **[SHOW]**를 누릅니다.

RPN 모드에서 각 프롬프트는 변수 값을 X 레지스터에 저장하고 스택 올리를 비활성화합니다. 프롬프트에 숫자를 입력하면 해당 숫자가 X 레지스터의 값을 대체합니다. **[R/S]**를 누르면 스택 올리가 활성화되므로 값이 스택에 저장됩니다.

---

## 방정식 구문

방정식은 계산 방법을 결정하는 특정 규칙을 따릅니다.

- 연산자 상호 작용 방법
- 방정식에 유효한 함수
- 방정식의 구문 오류가 확인되는 방법

## 연산자 우선순위

방정식의 연산자는 특정 순서대로 처리되므로 계산이 논리적이며 예측 가능합니다.

순서	연산	예
1	괄호	$(X+1)$
2	기능	$SIN(X+1)$
3	거듭제곱( $\square^{yx}$ )	$X^3$
4	음수( $\square/\square$ )	$-A$
5	곱하기 및 나누기	$X \times Y, A \div B$
6	더하기 및 빼기	$P+Q, A-B$
7	등식	$B=C$

따라서 예를 들어, 괄호 안의 모든 연산은 괄호 밖의 연산보다 먼저 수행됩니다.

예:

방정식	의미
$A \times B^3 = C$	$a \times (b^3) = c$
$(A \times B)^3 = C$	$(a \times b)^3 = c$
$A + B \div C = 12$	$a + (b/c) = 12$
$(A + B) \div C = 12$	$(a + b) / c = 12$
$\%CHG(T+12, A-6)^2$	$[\%CHG((t+12), (a-6))]^2$

## 방정식 함수

다음 표에서는 방정식에서 유효한 함수를 나열합니다. 부록 G, “연산 색인”에서도 이 정보를 제공합니다.

LN	LOG	EXP	ALOG	SQ	SQRT
INV	IP	FP	RND	ABS	!
SGN	INTG	IDIV	RMDR		
SIN	COS	TAN	ASIN	ACOS	ATAN
SINH	COSH	TANH	ASINH	ACOSH	ATANH
→DEG	→RAD	HMS→	→HMS	%CHG	XROOT
→L	→GAL	→MILE	→KM	nCr	nPr
→KG	→↴LB	→°C	→°F	→CM	→IN
SEED	ARG	RAND	$\pi$		
+	-	×	÷	^	
$s_x$	$s_y$	$\sigma_x$	$\sigma_y$	$\bar{x}$	$\bar{y}$
$\bar{x}_w$	$\hat{x}$	$\hat{y}$	$r$	$m$	$b$
$n$	$\Sigma x$	$\Sigma y$	$\Sigma x^2$	$\Sigma y^2$	$\Sigma xy$

편의상, 한두 개의 인수가 필요한 접두부 유형 함수는 입력 시 왼쪽 괄호를 표시합니다.

두 개의 인수가 필요한 접두부 함수는 %CHG, XROOT, IDIV, RMDR, nCr 및 nPr입니다. 두 인수를 쉼표로 구분합니다.

✓ 방정식에서 XROOT 함수는 RPN 사용과는 반대 순서로 인수를 사용합니다. 예를 들어,  $-8$  [ENTER] 3 [XROOT] 는 XROOT(3, -8)과 같습니다.

✓ 다른 두 인수 함수는 모두 RPN에 사용된 Y, X 순서대로 인수를 사용합니다. 예를 들어, 28 [ENTER] 4 [nCr] 는 nCr(28, 4)와 같습니다.

두 인수 함수에서 두 번째 인수가 음수인 경우 주의합니다. 다음은 올바른 방정식입니다.

## 6-16 방정식 입력 및 계산

%CHG(-X,-2)

%CHG(X,(-Y))

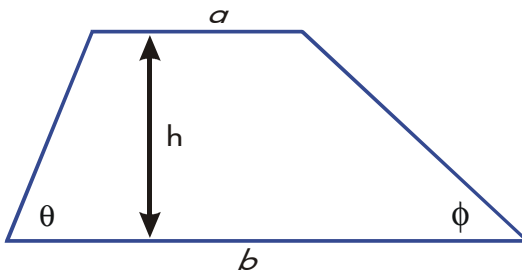
8개 방정식 함수의 이름은 그와 동등한 연산과는 다릅니다.

RPN 연산	방정식 함수
$x^2$	SQ
$\sqrt{x}$	SQRT
$e^x$	EXP
$10^x$	ALOG
$1/x$	INV
$\sqrt[y]{x}$	XROOT
$y^x$	^
INT÷	IDIV

### 예: 사다리꼴의 둘레

다음은 사다리꼴의 둘레를 계산하는 방정식입니다. 책에서는 방정식이 이런 방식으로 나타날 수 있습니다.

$$\text{둘레} = a + b + h \left( \frac{1}{\sin \theta} + \frac{1}{\sin \phi} \right)$$



다음은 HP 35s 방정식의 구문 규칙에 따른 방정식입니다.

항목 그룹화를 위해 괄호가 사용됨

$$P=A+B+H \times (1 \div \text{SIN}(T)+1 \div \text{SIN}(F))$$

단일 문자  
이름

명시적 곱하기  
(선택 사항)

나누기가 더하기보다  
먼저 수행됨

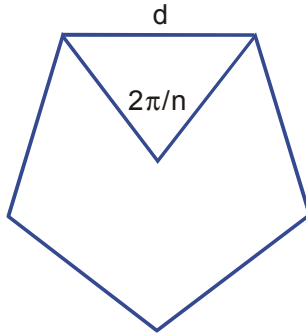
다음 방정식도 구문 규칙을 따릅니다. 이 방정식에서는 소수 형태인  $1 \div \text{SIN}(T)$  대신 역함수  $\text{INV}(\text{SIN}(T))$ 를 사용합니다. SIN 함수는 INV 함수 안에 “중첩”됩니다. INV는  $\sqrt{x}$ 에 의해 입력됩니다.

$$P=A+B+H \times (\text{INV}(\text{SIN}(T))+\text{INV}(\text{SIN}(F)))$$

### 예: 다각형의 넓이

길이가  $d$ 인  $n$ 개 변이 있는 정다각형의 넓이를 구하는 방정식은 다음과 같습니다.

$$\text{넓이} = \frac{1}{4} n d^2 \frac{\cos(\pi/n)}{\sin(\pi/n)}$$



이 방정식을 다음과 같이 지정할 수 있습니다.

$$A=0, 25 \times N \times D^2 \times \text{COS}(\pi \div N) \div \text{SIN}(\pi \div N)$$

연산자와 함수가 어떻게 결합되어 원하는 방정식을 제공하는지 확인합니다.

## 6-18 방정식 입력 및 계산

다음 키 입력을 사용하여 방정식 목록에 방정식을 입력할 수 있습니다.

**[EQN] [RCL] [A] [↵] [=] [·] [2] [5] [×] [RCL] [N] [×] [RCL] [D] [y<sup>x</sup>] [2] [×]**  
**[COS] [↵] [π] [÷] [RCL] [N] [>] [÷] [SIN] [↵] [π] [÷] [RCL] [N] [ENTER]**

## 구문 오류

이 계산기에서는 방정식을 계산할 때까지 방정식의 구문을 확인하지 않습니다. 오류가 탐지되면 **SYNTAX ERROR**가 표시되고 커서가 첫 번째 오류 위치에 표시됩니다. 방정식을 편집하여 오류를 수정해야 합니다. 이 장의 앞부분에 있는 “방정식 편집 및 지우기”를 참조하십시오.

계산 시까지 방정식 구문을 확인하지 않기 때문에 HP 35s를 사용하여 실제 메시지일 수 있는 “방정식”을 만들 수 있습니다. 이 기능은 13장에서 설명한 것처럼 프로그램에서 특히 유용합니다.

---

## 방정식 확인

방정식을 보고 있는 경우(방정식을 입력하고 있지 않음) **[↵] [SHOW]**를 눌러 방정식의 체크섬과 방정식 길이를 표시할 수 있습니다. 값을 디스플레이에 유지하려면 **[SHOW]** 키를 누르고 있습니다.

체크섬은 이 방정식을 고유하게 식별하는 4자리수 16진수 값입니다. 방정식을 잘못 입력하면 이 체크섬을 갖지 않습니다. 길이는 방정식에 사용된 계산기 메모리의 바이트 수입니다.

체크섬과 길이를 사용하여 입력한 방정식이 올바른지 확인할 수 있습니다. 예에서 입력한 방정식의 체크섬 및 길이가 이 설명서에 표시된 값과 일치해야 합니다.

### 예: 방정식의 체크섬 및 길이

이 장의 앞부분에 있는 파이프 부피 방정식의 체크섬과 길이를 구합니다.

키:

**EQN**

(필요한 경우 **^**)

**↵** **SHOW**

(누르고 있음)

(놓음)

**C**

디스플레이:

$$V=0.25 \times P \times D^2 \times L$$

$$CK=49CA$$

$$LN=14$$

$$V=0.25 \times P \times D^2 \times L$$

설명:

원하는 방정식을 표시합니다.

방정식의 체크섬과 길이를  
표시합니다.

방정식을 다시 표시합니다.  
방정식 모드를 종료합니다.



## 방정식 풀기

---

6장에서는 **[ENTER]**를 사용하여 지정 유형 방정식에서 좌변 변수의 값을 구할 수 있는 방법을 살펴보았습니다. SOLVE를 사용하여 모든 방정식 유형의 모든 변수 값을 구할 수 있습니다.

예를 들어, 다음 방정식의 경우

$$x^2 - 3y = 10$$

이 방정식에서  $y$  값을 알고 있으면 SOLVE를 사용하여 미지수  $x$ 의 값을 구할 수 있습니다.  $x$  값을 알고 있으면 SOLVE를 사용하여 미지수  $y$ 의 값을 구할 수 있습니다. “단어 문제”의 경우도 마찬가지입니다.

$$\text{이윤} \times \text{원가} = \text{가격}$$

세 변수 중 두 개를 알고 있으면 SOLVE를 사용하여 세 번째 변수 값을 계산할 수 있습니다.

방정식에 변수가 하나뿐이거나 한 변수를 제외한 모든 변수에 대해 알려진 값이 제공되는 경우  $x$  값을 구하려면 방정식의 근을 구합니다. 방정식의 근은 등식 또는 지정 방정식이 정확하게 균형을 이루거나 수식 방정식이 0과 같을 때 발생합니다.

---

## 방정식 풀기

### 기본 제공 방정식을 제외한 미지수 변수 방정식을 풀려면

1. **[EQN]**을 누르고 원하는 방정식을 표시합니다. 필요한 경우 6장의 “방정식 목록에 방정식 입력”에 설명된 대로 방정식을 입력합니다.

2. **[F2] SOLVE**를 누른 다음 미지수 변수에 대한 키를 누릅니다. 예를 들어,  $x$ 의 값을 구하려면 **[F2] SOLVE X**를 누릅니다. 그러면 방정식의 다른 모든 변수 값을 묻는 프롬프트가 표시됩니다.
  3. 각 프롬프트에 대해 원하는 값을 입력합니다.
    - 표시되는 값이 원하는 값인 경우 **[R/S]**를 누릅니다.
    - 다른 값을 원하는 경우 값을 입력하거나 계산하고 **[R/S]**를 누릅니다. 자세한 내용은 6장의 “방정식 프롬프트에 응답”을 참조하십시오.
- [C]** 또는 **[R/S]**를 눌러 실행 중인 계산을 중지할 수 있습니다.

근을 구하면 근이 관계 변수에 저장되고 변수 값이 디스플레이에 표시됩니다. 또한 X 레지스터에는 근이 포함되고, Y 레지스터에는 이전 추정 값이나 0이 포함되고, Z 레지스터에는 근 D 값(0이어야 함)이 포함됩니다.

일부 복잡한 수학적 조건에서는 명확한 해를 찾을 수 없으며 계산기에 **NO ROOT FOUND**가 표시됩니다. 이 장의 뒷부분에 나오는 “결과 확인”과 부록 D의 “결과 해석” 및 “SOLVE를 사용하여 근을 구할 수 없는 경우”를 참조하십시오.

특정 방정식에서는 방정식을 풀기 전에 미지수 변수에 대해 한두 개의 초기 추정 값을 제공하는 것이 도움이 됩니다. 이렇게 하면 계산 속도가 빨라지며, 결과를 실제 해로 유도하고, 여러 개의 해를 구할 수 있습니다(해당되는 경우). 이 장의 뒷부분에 있는 “SOLVE에 대한 초기 추정값 선택”을 참조하십시오.

### 예: 선형 운동의 방정식 풀기

자유낙하 개체의 운동 방정식은 다음과 같습니다.

$$d = v_0 t + \frac{1}{2} g t^2$$

여기서  $d$ 는 거리,  $v_0$ 은 초기 속도,  $t$ 는 시간,  $g$ 는 중력으로 인한 가속도입니다.

다음과 같이 방정식을 입력합니다.

## 7-2 방정식 풀기

키:	디스플레이:	설명:
<b>[▶] [CLEAR] [3] (3ALL)</b>		메모리를 지웁니다.
<b>[&lt;] (Y) [ENTER]</b>		
<b>[EQN]</b>	3*3 lin. solve EQN LIST TOP	방정식 모드를 선택합니다.
<b>[RCL] [D] [↵] [=] [RCL]</b>		방정식을 시작합니다.
<b>[V] [X] [RCL] [T] [+]</b>	$D=V \times T + \_$	
<b>[.] [5] [X] [RCL] [G] [X]</b>	$\leftarrow =V \times T + 0.5 \times G \times T^2 \_$	
<b>[RCL] [T] [<math>y^x</math>] [2]</b>		
<b>[ENTER]</b>	$D=V \times T + 0.5 \times G \times T^2$	방정식을 종료하고 왼쪽 끝을 표시합니다.
<b>[↵] [SHOW]</b>	CK=FB3C LN=15	체크섬 및 길이입니다.

$g$ (중력으로 인한 가속도)는 변수로 포함되므로 다른 단위에 맞게 변경할 수 있습니다( $9.8\text{m/s}^2$  또는  $32.2\text{ft/s}^2$ ).

개체가 멈춘 상태에서 시작하여 5초 내에 몇 미터나 떨어지는지 계산합니다. 방정식 모드가 켜져 있고 원하는 방정식이 이미 디스플레이에 있으므로  $D$ 의 값을 구할 수 있습니다.

키:	디스플레이:	설명:
<b>[▶] [SOLVE]</b>	SOLVE_	미지수 변수를 묻는 프롬프트를 표시합니다.
<b>[D]</b>	V? <i>값</i>	$D$ 를 선택하고 $V$ 를 묻는 프롬프트를 표시합니다.
<b>[0] [R/S]</b>	T? <i>값</i>	0을 $W$ 에 저장하고 $T$ 를 묻는 프롬프트를 표시합니다.
<b>[5] [R/S]</b>	G? <i>값</i>	5를 $G$ 에 저장하고 $G$ 를 묻는 프롬프트를 표시합니다.
<b>[9] [.] [8] [R/S]</b>	SOLVING D= 122.5000	9.8을 $G$ 에 저장하고 $D$ 의 값을 구합니다.

동일한 방정식을 사용하여 다른 계산을 시도합니다. 개체가 멈춘 상태에서 500미터를 떨어지는 데 걸리는 시간은 얼마입니까?

키:

**EQN**  
**→ SOLVE T**  
**5 0 0 R/S**  
**R/S**  
**R/S**

디스플레이:

$D = V \times T + 0.5 \times G \times T^2$   
 D?  
 122.5  
 V?  
 0  
 G?  
 9.8  
 SOLVING  
 T=  
 10.1015

설명:

방정식을 표시합니다.  
 T의 값을 구하고 D를 묻는 프롬프트를 표시합니다.  
 500을 D에 저장하고 V를 묻는 프롬프트를 표시합니다.  
 0을 V에 유지하고 G를 묻는 프롬프트를 표시합니다.  
 9.8을 G에 유지하고 T의 값을 구합니다.

### 예: 이상 기체 법칙 방정식 풀기

이상 기체 법칙에서는 이상 기체의 압력, 부피, 온도 및 양(몰) 간 관계에 대해 설명합니다.

$$P \times V = N \times R \times T$$

여기서 P는 압력(기압 또는 N/m<sup>2</sup>), V는 부피(리터), N은 기체 몰 수, R은 보편 기체 상수(0.0821리터-기압/몰-K 또는 8.314J/몰-K), T는 온도(켈빈 온도: K=°C + 273.1)입니다.

다음과 같이 방정식을 입력합니다.

키:

**EQN RCL P X**  
**RCL V ↵ =**  
**RCL N X**  
**RCL R X RCL T**  
**ENTER**  
**↵ SHOW**

디스플레이:

P×\_  
 P×V=N×R×T\_  
 P×V=N×R×T  
 CK=EDC8  
 LN=9

설명:

방정식 모드를 선택하고 방정식을 시작합니다.  
 방정식을 종료하고 표시합니다.  
 체크섬 및 길이입니다.

24°C에서 2리터 병에 이산화탄소 기체 0.005몰이 들어 있습니다. 기체가 이상적인 기체 운동을 한다고 가정하여 기체의 압력을 계산합니다. 방정식 모드가 켜져 있고 원하는 방정식이 이미 디스플레이에 있으므로 P의 값을 구할 수 있습니다.

키:	디스플레이:	설명:
SOLVE P	V? 값	P의 값을 구하고 V를 묻는 프롬프트를 표시합니다.
2 R/S	N? 값	2를 V에 저장하고 N을 묻는 프롬프트를 표시합니다.
. 0 0 5 R/S	R? 값	.005를 N에 저장하고 R을 묻는 프롬프트를 표시합니다.
. 0 8 2 1 R/S	T? 값	.0821을 R에 저장하고 T를 묻는 프롬프트를 표시합니다.
2 4 + 2 7 3 .	T? 297.1000	T(켈빈 온도)를 계산합니다.
1 ENTER	SOLVING	297.1을 T에 저장하고 P(압)의 값을 구합니다.
R/S	P= 0.0610	

5리터 플라스크에 질소 기체가 들어 있습니다. 온도가 18°C일 때 압력은 0.05기압입니다. 기체의 밀도를 계산합니다( $N \times 28 / V$ , 여기서 28은 질소의 분자량).

키:	디스플레이:	설명:
EQN	$P \times V = N \times R \times T$	방정식을 표시합니다.
SOLVE N	P? 0.0610	N의 값을 구하고 P를 묻는 프롬프트를 표시합니다.
. 0 5 R/S	V? 2.0000	.05를 P에 저장하고 V를 묻는 프롬프트를 표시합니다.
5 R/S	R? 0.0821	5를 V에 저장하고 R을 묻는 프롬프트를 표시합니다.
R/S	T? 297.1000	이전 R을 유지하고 T를 묻는 프롬프트를 표시합니다.
✓ 1 8 ENTER 2 7 3	T? 291.1000	T(켈빈 온도)를 계산합니다.
. 1 +		

	<b>R/S</b>	SOLVING	291.1을 $\pi$ 에 저장하고 $N$ 의 값을 구합니다.
		N=	
✓	<b>2</b> <b>8</b> <b>X</b>	0.0105	
		0.2929	질량(그램)인 $N \times 28$ 을 계산합니다.
✓	<b>RCL</b> <b>V</b> <b>÷</b>	0.0586	밀도(리터당 그램)를 계산합니다.

## 기본 제공 방정식 풀기

기본 제공 방정식은 “2\*2 lin. solve” ( $Ax+By=C, Dx+Ey=F$ ) 및 “3\*3 lin. solve” ( $Ax+By+Cz=D, Ex+Fy+Gz=H, Ix+Jy+Kz=L$ )입니다. 두 방정식 중 하나를 선택하면 **XEQ**, **ENTER** 및 **/** 키는 작동하지 않습니다. **SOLVE**를 누르면 2\*2 케이스에 대해 6개 변수(A-F)가 요청되거나 3\*3 케이스에 대해 12개 변수(A-L)가 요청되며, 이러한 변수를 사용하여 2\*2 선형 방정식 체계의  $x, y$  또는 3\*3 선형 방정식 체계의  $x, y$  및  $z$ 를 구합니다. 결과는 변수  $x, y$  및  $z$ 에 저장됩니다. 계산기는 무한히 많은 해를 가지고 있거나 해가 없는 케이스를 탐지할 수 있습니다.

예: 연립 방정식  $\begin{cases} x+2y=5 \\ 3x+4y=11 \end{cases}$  에서  $x, y$ 의 값 구하기

키:	디스플레이:	설명:
<b>EQN</b>	3*3 lin. solve	방정식 모드를 시작합니다.
<b>∨</b>	EQN LIST TOP	
<b>SOLVE</b>	EQN LIST TOP	기본 제공 방정식을 표시합니다.
	2*2 lin. solve	
	A?	A를 묻는 프롬프트를 표시합니다.
	값	
<b>1</b> <b>R/S</b>	B?	1을 A에 저장하고 B를 묻는 프롬프트를 표시합니다.
	값	
<b>2</b> <b>R/S</b>	C?	2를 B에 저장하고 C를 묻는 프롬프트를 표시합니다.
	값	
<b>5</b> <b>R/S</b>	D?	5를 C에 저장하고 D를 묻는 프롬프트를 표시합니다.
	값	
<b>3</b> <b>R/S</b>	E?	3을 D에 저장하고 E를 묻는 프롬프트를 표시합니다.
	값	

## 7-6 방정식 풀기

**4** **R/S**

F?

4를 F에 저장하고 F를 묻는 프롬프트를 표시합니다.

값

**1** **1** **R/S**

X=



11을 F에 저장하고 x와 y를 계산합니다.

1.0000



**∇**

y=



y의 값입니다.

2.0000



## SOLVE 이해 및 제어

SOLVE는 먼저 미지수 변수에 대해 직접 방정식을 풀려고 합니다. 이 시도가 실패하면 SOLVE는 반복 절차로 변경합니다. 이 절차에서는 먼저 미지수 변수에 대한 두 개의 초기 추정값을 사용하여 방정식을 계산합니다. 두 추정값의 결과를 기반으로 SOLVE는 더 나은 다른 추정값을 생성합니다. 연속적인 반복을 통해 SOLVE를 방정식의 값을 0과 같게 만드는 미지수의 값을 구합니다.

SOLVE는 **[XEQ]**와 동일한 방식으로 방정식을 계산합니다. 방정식의 모든 "="는 "-"로 처리됩니다. 예를 들어, 이상 기체 법칙 방정식은  $P \times V - (N \times R \times T)$ 로 계산됩니다. 이렇게 하면 등식 또는 지정 방정식은 근에서 균형을 이루고, 수식 방정식은 근에서 0과 같습니다.

일부 방정식은 다른 방정식보다 더 풀기 어렵습니다. 경우에 따라 해를 구하기 위해 초기 추정값을 입력해야 합니다. 아래의 "SOLVE에 대한 초기 추정값 선택"을 참조하십시오. SOLVE를 사용하여 해를 구할 수 없는 경우 계산기에 **NO ROOT FND**가 표시됩니다.

SOLVE 작동 방식에 대한 자세한 내용은 부록 D를 참조하십시오.

## 결과 확인

SOLVE 계산이 끝난 후 스택에 남은 값을 검토하여 결과가 실제 방정식의 해인지 확인할 수 있습니다.

- X 레지스터(표시된 변수를 지우려면 **[C]**를 누름)에는 미지수의 해(근), 즉 방정식 계산을 0과 같게 만드는 값이 들어 있습니다.

- ✓ ■ Y 레지스터(**R1** 누름)는 근의 이전 추정값을 포함하거나 0과 같습니다. 이 숫자는 X 레지스터의 값과 같아야 합니다. 그렇지 않으면 반환된 근은 근사값일 뿐이며 X 레지스터와 Y 레지스터의 값에서 근이 괄호 안에 표시됩니다. 이러한 괄호 안의 숫자는 서로 가까워야 합니다.
- ✓ ■ Z 레지스터(다시 **R1** 누름)에는 근에서 계산된 방정식의 D 값이 포함됩니다. 정확한 근에서는 값이 0이어야 합니다. 0이 아니면 주어진 근은 근사값일 뿐이며, 이 숫자는 0과 가까워야 합니다.

계산이 **NO ROOT FND**로 끝나는 경우 계산기에서 근을 수렴할 수 없습니다.

**→gd|** 또는 **←**를 눌러 메시지를 지우면 근의 최종 추정값인 X 레지스터의 값을 확인할 수 있습니다. X 레지스터와 Y 레지스터의 값에서 근을 구하기 위해 마지막으로 검색된 간격이 괄호 안에 표시됩니다. Z 레지스터에는 근의 최종 추정값에서 계산된 방정식의 값이 포함됩니다.

- X 레지스터와 Y 레지스터의 값이 서로 가깝지 않거나 Z 레지스터 값이 0에 가깝지 않은 경우 X 레지스터의 추정값은 근이 아닐 가능성이 큼니다.
- X 레지스터와 Y 레지스터의 값이 서로 *가까우며*, Z 레지스터 값이 0에 *가까운* 경우 X 레지스터의 추정값이 근의 근사값일 수 있습니다.

## SOLVE 계산 일시 중단

계산을 일시 중지하려면 **C** 또는 **R/S**를 누릅니다. “**INTERRUPTED**”라는 메시지가 표시됩니다. 근의 현재 최적 추정값이 미지수 변수에 있습니다. 스택을 방해하지 않고 해당 값을 보려면 **←** **VIEW**를 사용합니다. 그러나 값을 다시 구할 수는 없습니다.

## SOLVE에 대한 초기 추정값 선택

두 개의 초기 추정값은 다음에서 가져옵니다.

- 현재 미지수 변수에 저장된 숫자
- X 레지스터(디스플레이)의 숫자



이러한 소스는 추정값 입력 여부에 관계없이 추정값에 사용됩니다. 하나의 추정값만 입력하고 변수에 저장하면 방금 변수에 저장한 숫자가 디스플레이에 유지되므로 두 번째 추정값은 동일한 값이 됩니다. 이 경우 계산기에서 두 개의 다른 추정값이 있도록 하나의 추정값을 약간 변경합니다.

고유한 추정값을 입력하면 다음과 같은 이점이 있습니다.

- 검색 범위를 좁혀 추정값을 통해 해를 구하는 데 걸리는 시간을 단축시킬 수 있습니다.
- 수학적 해가 두 개 이상 있는 경우 추정값을 통해 SOLVE 절차를 원하는 결과나 결과 범위로 유도할 수 있습니다. 예를 들어, 다음과 같은 선형 운동 방정식이 있습니다.

$$d = v_0 t + 1/2 g t^2$$

이 방정식에서  $t$ 의 해는 두 개가 될 수 있습니다. 적절한 추정값을 입력하여 결과를 필요한 해로 유도할 수 있습니다.

이 장 앞부분의 이 방정식을 사용한 예에서는 해당 예의 첫 부분에서  $t$ 의 값을 저장하고  $D$ 의 값을 구했기 때문에  $t$ 의 값을 구하기 전에 추정값을 입력할 필요가 없었습니다.  $t$ 에 남은 값이 양호한(실제) 값이었으므로  $t$ 의 값을 구했을 때 추정값으로 사용되었습니다.

- 방정식에서 미지수에 대해 특정 값을 허용하지 않는 경우 추정값을 통해 해당 값이 발생하지 않도록 할 수 있습니다. 예를 들어, 다음과 같은 방정식이 있습니다.

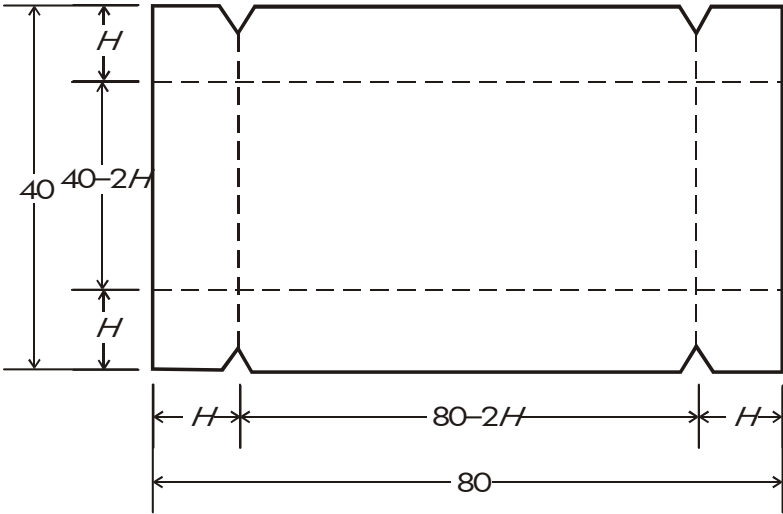
$$y = t + \log x$$

$x \leq 0$ 일 경우 이 방정식에서 오류가 발생합니다(NO ROOT FND라는 메시지가 표시됨).

다음 예에서는 방정식이 여러 개의 근을 갖지만 추정값을 통해 원하는 근을 찾을 수 있습니다.

**예: 추정값을 사용하여 근 찾기**

40cm X 80cm의 직사각형 금속판을 사용하여 부피가 7500cm<sup>3</sup>이고 위쪽이 열린 상자를 만듭니다. 지정된 부피를 제공하는 상자의 높이(즉, 네 변에서 각각 접어 올린 양)를 구해야 합니다. *짧은* 상자보다 *긴* 상자가 선호됩니다.



$H$ 가 높이이면 상자의 길이는  $(80 - 2H)$ 이고 너비는  $(40 - 2H)$ 입니다. 부피  $V$ 는 다음과 같습니다.

$$V = (80 - 2H) \times (40 - 2H) \times H$$

이 방정식을 단순화하여 다음과 같이 입력할 수 있습니다.

$$V = (40 - H) \times (20 - H) \times 4 \times H$$

다음과 같이 방정식을 입력합니다.

**키:**

**디스플레이:**

**설명:**

[EQN] [RCL] [V] [↩] [=]

$V =$  \_

방정식 모드를 선택하고 방정식을 시작합니다.

[()] [4] [0] [-] [RCL] [H] [>]

$V = (40 - H)$  \_

**7-10 방정식 풀기**

X ( ) 2 0 -  
 RCL H >  
 X 4 X RCL H  
 ENTER  
 SHOW

$(40-H) \times (20-H)$   
 $H) \times (20-H) \times 4 \times H$   
 $V = (40-H) \times (20-H)$   
 $CK = 49A4$   
 $LN = 19$

방정식을 종료하고 표시합니다.  
 체크섬 및 길이입니다.

원하는 부피를 제공하는 길고 좁은 상자나 짧고 편평한 상자를 만드는 것이 모두 가능할 것 같습니다. 긴 상자가 선호되기 때문에 높이의 초기 추정값이 더 커지는 것도 당연합니다. 그러나 금속판의 너비가 40cm밖에 되지 않으므로 20cm보다 큰 높이는 물리적으로 불가능합니다. 따라서 초기 추정값은 10과 20이 적합합니다.

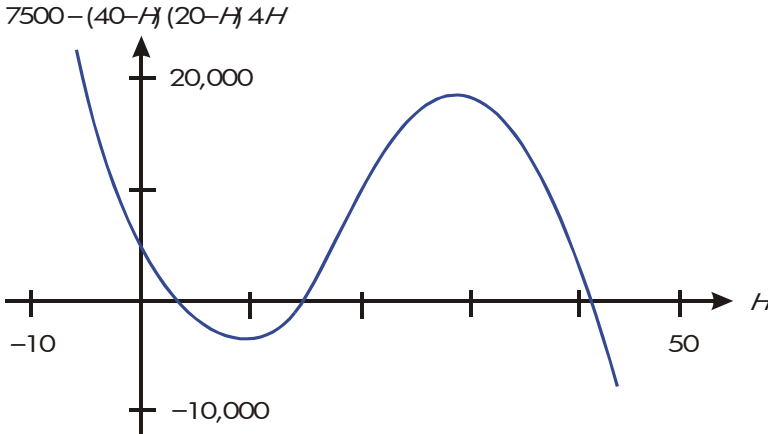
키:	디스플레이:	설명:
C		방정식 모드를 종료합니다.
1 0 → STO H		추정 하한값과 상한값을 저장합니다.
ENTER 2 0	20	
EQN	$V = (40-H) \times (20-H)$	현재 방정식을 표시합니다.
→ SOLVE H	V?	H의 값을 구하고 V를 묻는 프롬프트를 표시합니다.
	값	
7 5 0 0 R/S	H=	7500을 V에 저장하고 H의 값을 구합니다.
	15.0000	

이제 근의 이전 추정값(Y 레지스터) 및 근에서 계산된 방정식의 값(Z 레지스터)을 검토하여 이 해의 품질, 즉 정확한 근의 반환 여부를 확인합니다.

	키:	디스플레이:	설명:
✓	R↓	15.0000	Y 레지스터의 이 값은 최종 결과 직전의 추정값입니다. 이 값이 해와 같으므로 해는 정확한 근입니다.
✓	R↓	0.0000	Z 레지스터의 이 값은 근에서 방정식이 0과 같음을 표시합니다.

원하는 상자의 치수는  $50 \times 10 \times 15\text{cm}$ 입니다. 높이의 상한값(20cm)을 무시하고 초기 추정값을 30cm와 40cm로 사용했다면 물리적으로 의미가 없는 근인 42.0256cm의 높이를 얻었을 것입니다. 0cm와 10cm의 작은 초기 추정값을 사용했다면 짧고 편평한 상자를 만드는 2.9774cm의 높이를 얻었을 것입니다.

어떤 추정값을 사용할지 잘 모르겠으면 방정식의 작동 방식을 이해하는 데 유용한 그래프를 사용할 수 있습니다. 미지수의 여러 값에 대해 방정식을 계산합니다. 그래프의 각 점에서 방정식을 표시하고 **[XEQ]**를 누릅니다.  $x$ 를 묻는 프롬프트가 표시되면  $x$  좌표를 입력합니다. 그러면 방정식의 해당 값인  $y$  좌표를 구할 것입니다. 위의 문제에서는 항상  $V = 7500$ 을 설정하고  $H$ 의 값을 변경하여 방정식의 다른 값을 구합니다. 이 방정식의 값은 방정식에서 좌변과 우변의 차와 같습니다. 이 방정식의 값을 도면으로 작성하면 다음과 같이 나타납니다.



## 추가 정보

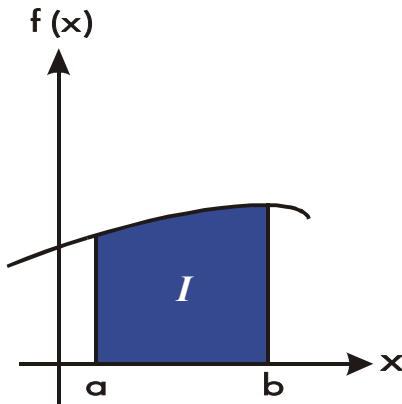
이 장에서는 다양한 응용 프로그램에서 근 또는 미지수를 구하는 방법에 대한 지침을 제공합니다. 부록 D에는 SOLVE 알고리즘의 작동 방식, 결과 해석 방법, 해를 찾지 못한 경우 수행되는 동작 및 잘못된 결과가 발생할 수 있는 조건에 대한 자세한 내용이 들어 있습니다.

## 7-12 방정식 풀기

## 방정식 적분

수학, 과학 및 공학 문제 계산에서는 대부분 함수의 정적분이 필요합니다. 함수가  $f(x)$ 로 표시되고 적분 간격이  $a$ 에서  $b$ 이면 수학적으로 적분을 다음과 같이 표현할 수 있습니다.

$$I = \int_a^b f(x) dx$$



수량  $I$ 는 기하학적으로 함수  $f(x)$ 의 그래프,  $x$ 축, 한계  $x=a$  및  $x=b$ 로 경계가 지워진 영역의 넓이로 해석할 수 있습니다( $f(x)$ 가 적분 간격 전체에서 음수가 아닌 경우).

연산( $\int$  FN)은 지정된 변수( $\int$  FN d\_)와 관련해서 현재 방정식을 적분합니다. 함수에 둘 이상의 변수가 있을 수 있습니다.

# 방정식 적분( ∫ FN)

## 방정식을 적분하려면

1. 피적분 함수를 정의하는 방정식이 방정식 목록에 저장되어 있지 않으면 입력하고(6장의 “방정식 목록에 방정식 입력” 참조) 방정식 모드를 종료합니다. 일반적으로 방정식에는 수식만 포함됩니다.
- ✓ 2. 적분 한계 입력: *하한값* 을 입력하고 **ENTER** 를 누른 다음 *상한값* 을 입력합니다.
3. 방정식 표시: **EQN** 을 누르고 필요한 경우 **^** 또는 **∇** )를 눌러 방정식 목록을 스크롤하여 원하는 방정식을 표시합니다.
4. 적분 변수 선택: **←** **∫** 변수를 누릅니다. 계산이 시작됩니다.

**∫** 는 계산기의 다른 모든 연산보다 훨씬 많은 메모리를 사용합니다. **∫** 실행으로 인해 **MEMORY FULL**이라는 메시지가 표시되는 경우 부록 B를 참조하십시오.

**C** 또는 **R/S** 를 눌러 실행 중인 적분 계산을 중지할 수 있으며, “**INTERRUPTED**”라는 메시지가 줄 2에 표시됩니다. 그러나 적분을 다시 시작할 수는 없습니다. 계산이 정상적으로 완료될 때까지 적분에 대한 정보를 사용할 수 없습니다.

디스플레이 형식 설정은 함수에 대해 가정되고 결과에 사용되는 정확도 수준에 영향을 줍니다. **ALL** 및 높은 **FIX**, **SCI**, **ENG** 설정에서는 적분이 보다 정확하지만 *훨씬* 오래 걸립니다. 결과의 *불확정*도는 Y 레지스터에 저장되어 적분 한계를 T 및 Z 레지스터로 올립니다. 자세한 내용은 이 장의 뒷부분에 있는 “적분의 정확도”를 참조하십시오.

## 다른 정보를 사용하여 동일한 방정식을 적분하려면

- ✓ 동일한 적분 한계를 사용하는 경우 **RT** **RT** 을 눌러 X 및 Y 레지스터로 이동합니다. 그런 다음 위의 목록에 있는 3단계에서 시작합니다. 다른 한계를 사용하려는 경우 2단계에서 시작합니다.

다른 방정식을 사용하여 다른 문제를 해결하려면 피적분 함수를 정의하는 방정식으로 1단계부터 새로 시작합니다.

**예: 베셀 함수**





















제1종 베셀 함수는 다음과 같이 표현할 수 있습니다.

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin t) dt$$










x 값 2와 3에 대한 베셀 함수를 구합니다.

다음 피적분 함수를 정의하는 수식을 입력합니다.

$$\cos(x \sin t)$$

키:	디스플레이:	설명:
  		메모리를 지웁니다.
  (Y) 		
	3*3 lin. solve EQN LIST TOP	방정식 모드를 선택합니다.
  	COS(X <sub>2</sub> )	방정식을 입력합니다.
 	COS(X×SIN( <sub>2</sub> ))	
 	COS(X×SIN(T <sub>2</sub> ))	
 	COS(X×SIN(T <sub>2</sub> ))_	
	COS(X×SIN(T <sub>2</sub> ))	수식을 종료하고 왼쪽 끝을 표시합니다.
 	CK=E1EC LN=13	체크섬 및 길이입니다.
		방정식 모드를 종료합니다.

이제 0에서 π ; x = 2까지 t와 관련해서 이 함수를 적분합니다.

키:	디스플레이:	설명:
  (2RAD)		라디안 모드를 선택합니다.
   	3.1416	적분 한계를 입력합니다(먼저 하한값 입력).
	COS(X×SIN(T))	함수를 표시합니다.
 	∫ FN d_	적분 변수를 묻는 프롬프트를 표시합니다.

**T** X? X의 값을 묻는 프롬프트를 표시합니다.

**2** **R/S** INTEGRATING x = 2. 적분을 시작하고 다음 방정식의 결과를 계산합니다.

∫ =  
0.7034  
 $\int_0^{\pi} f(t)$

✓ **↵** **π** **÷** 0.2239 J(2)의 최종 결과입니다.

이제 동일한 적분 한계를 사용하여 J(3)을 계산합니다. 이후의 π로 나누기에 의해 스택에서 삭제되었으므로 적분 한계(0, π)를 다시 지정해야 합니다.

**키:**

**디스플레이:**

**설명:**

✓ **0** **ENTER** **↵** **π** 3.1416 적분 한계를 입력합니다(먼저 하한값 입력).

**EQN** COS(X×SIN(T)) 현재 방정식을 표시합니다.

**↵** **/** ∫ FN d\_ 적분 변수를 묻는 프롬프트를 표시합니다.

**T** X? X의 값을 묻는 프롬프트를 표시합니다.

2.0000

**3** **R/S** INTEGRATING x = 3. 적분을 시작하고 다음 방정식의 결과를 계산합니다.

∫ =  
-0.8170  
 $\int_0^{\pi} f(t)$

✓ **↵** **π** **÷** -0.2601 J(3)의 최종 결과입니다.

✓ **예: 사인 적분**

통신 이론의 특정 문제(예: 이상적인 네트워크를 통한 펄스 전송)에서는 다음과 같은 형태의 적분(사인 적분이라고도 함)을 계산해야 합니다.

$$S_i(f) = \int_0^f \left( \frac{\sin x}{x} \right) dx$$

S<sub>i</sub>(2)를 구합니다.



다음 피적분 함수를 정의하는 수식을 입력합니다.

$$\frac{\sin x}{x}$$

계산기가 적분 하한값인  $x=0$ 에서 이 함수를 계산하려고 하면 오류(DIVIDE BY 0)가 발생합니다. 그러나 적분 간격의 끝점이 서로 매우 가깝거나 동일한 점의 개수가 매우 큰 경우가 아니면 적분 알고리즘은 일반적으로 적분 한계에서 함수를 계산하지 않습니다.

키:	디스플레이:	설명:
<b>EQN</b>	3*3 lin. solve EQN LIST TOP	방정식 모드를 선택합니다.
<b>SIN</b> <b>RCL</b> <b>X</b>	SIN(X)_	방정식을 시작합니다.
<b>&gt;</b>	SIN(X)_	이 경우 닫는 오른쪽 괄호가 필요합니다.
<b>÷</b> <b>RCL</b> <b>X</b>	SIN(X)÷X_	
<b>ENTER</b>	SIN(X)÷X	방정식을 종료합니다.
<b>↵</b> <b>SHOW</b>	CK=0EE0 LN=8	체크섬 및 길이입니다.
<b>C</b>		방정식 모드를 종료합니다.

이제 0에서  $2(\pi=2)$ 까지  $x$ (즉,  $\pi$ )와 관련해서 이 함수를 적분합니다.

키:	디스플레이:	설명:
<b>MODE</b> <b>2</b> (2RAD)		라디안 모드를 선택합니다.
<b>0</b> <b>STO</b> <b>X</b> <b>ENTER</b>	2_	적분 한계를 입력합니다(먼저 하한값 입력).
<b>2</b>		
<b>EQN</b>	SIN(X)÷X	현재 방정식을 표시합니다.
<b>↵</b> <b>∫</b> <b>X</b>	INTEGRATING ∫= 1.6054	$S(2)$ 의 결과를 계산합니다.

---

## 적분의 정확도

계산기는 적분 값을 정확하게 계산할 수 없으므로 근사값을 구합니다. 이 근사값의 정확도는 방정식에 의해 계산된 피적분 함수 자체의 정확도에 따라 달라집니다. 계산기의 버림 오류와 경험적 상수의 정확도에 의해 영향을 받습니다.

스파이크나 매우 급속한 진동과 같은 특정 특성을 가진 함수의 적분은 부정확하게 계산될 수도 있지만 그 가능성은 매우 적습니다. 문제가 발생할 수 있는 함수의 일반적인 특성과 이러한 문제를 처리하는 기술에 대해서는 부록 E에서 설명합니다.

## 정확도 지정

디스플레이 형식의 설정(FIX, SCI, ENG 또는 ALL)에 따라 적분 계산의 정밀도가 결정됩니다. 표시되는 자릿수가 클수록 계산된 적분의 정밀도가 커지고 계산 시간이 길어집니다. 표시되는 자릿수가 적을수록 계산 속도는 빨라지지만 계산기에서 지정된 자릿수까지만 함수가 정확하다고 가정합니다.

적분의 정확도를 지정하려면 디스플레이에 표시되는 자릿수가 피적분 함수의 값에서 정확하다고 간주되는 자릿수보다 크지 않도록 디스플레이 형식을 설정합니다. 동일한 수준의 정확도와 정밀도가 적분 결과에도 반영됩니다.

분수 표시 모드를 켜면(플래그 7 설정) 이전 디스플레이 형식에 의해 정확도가 지정됩니다.

## 정확도 해석

계산기는 적분을 계산한 후 해당 피적분 함수 결과의 예상 불확정도를 Y 레지스터에 저장합니다. 불확정도의 값을 보려면  $\boxed{x \leftrightarrow y}$  을 누릅니다.

예를 들어, 적분  $S_{i(2)}$ 가  $1.6054 \pm 0.0002$ 이면 0.0002가 해당 불확정도입니다.

### 예: 정확도 지정

디스플레이 형식을 SCI 2로 설정하고 이전 예의  $S_i/2$ 에 대한 수식에서 적분을 계산합니다.

키:	디스플레이:	설명:
<b>DISPLAY</b> <b>2</b> <b>(2SCI) 2</b>	1.61E0	소수 두 자릿수의 과학 표기법을 설정하고 함수가 소수 두 자릿수까지 정확하도록 지정합니다.
<b>R1</b> <b>R1</b>	0.00E0 2.00E0	Z 및 T 레지스터에서 X 및 Y 레지스터로 적분 한계를 롤다운합니다.
<b>EQN</b>	SIN(X)+X	현재 방정식을 표시합니다.
<b>X</b>	INTEGRATING ∫ = 1.61E0	소수 두 자릿수까지 적분 근사값을 구합니다.
<b>x↔y</b>	1.61E-2	적분 근사값의 불확정도입니다.

적분은  $1.61 \pm 0.0161$ 입니다. 소수 세 자릿수 이전에는 불확정도가 근사값에 영향을 주지 않으므로 이 근사값에 표시된 자릿수는 모두 정확하다고 간주할 수 있습니다.

근사값의 불확정도가 선택한 허용 값보다 크면 디스플레이 형식의 자릿수를 늘리고 적분을 반복할 수 있습니다( $f(x)$ 가 디스플레이에 표시된 자릿수까지 정확하게 계산되는 경우). 일반적으로 적분 계산의 불확정도는 디스플레이 형식에 지정된 자릿수가 하나씩 증가할 때마다 10의 인수만큼 감소합니다.

### 예: 정확도 변경

방금 계산한  $S_i/2$ 의 적분에 대해 결과가 소수 두 자릿수가 아닌 소수 네 자릿수까지 정확하도록 지정합니다.

키:

**[←] [DISPLAY] [2]**  
**(2SCI) [4]**

✓ **[R↓] [R↓]**

**[EQN]**

**[←] [ / ] X**

**[X↔Y]**

**[DISPLAY] [1] (2SCI) [4]**

**[MODE] [1] (1DEG)**

디스플레이:

1.6079E-2

0.0000E0

2.0000E0

SIN(X)÷X

INTEGRATING

∫ =

1.6054E0

1.6056E-4

0.0002

0.0002

설명:

정확도를 소수 네 자릿수로 지정합니다. 마지막 예의 불확정도가 여전히 디스플레이에 표시됩니다.

Z 및 T 레지스터에서 X 및 Y 레지스터로 적분 한계를 롤다운합니다.

현재 방정식을 표시합니다.

결과를 계산합니다.

불확정도가 이전에 계산한 SCI 2 불확정도의 약 1/100입니다.

FIX 4 형식으로 복원합니다.

도 모드로 복원합니다.

이 불확정도는 결과가 소수 세 자릿수까지만 정확할 수 있음을 나타냅니다. 실제로 이 결과는 이 적분의 실제 값과 비교할 때 소수 7자릿수까지 정확합니다. 결과의 불확정도는 보수적으로 계산되기 때문에 *계산기의 근사값은 대체로 불확정도가 나타내는 것보다 정확합니다.*

---

## 추가 정보

이 장에서는 다양한 응용 프로그램에서 HP 35s의 적분을 사용하는 방법에 대한 지침을 제공합니다. 부록 E에는 적분 알고리즘의 작동 방식, 잘못된 결과가 발생할 수 있는 조건 및 계산 시간이 길어지는 조건, 적분의 현재 근사값 확인에 대한 자세한 내용이 들어 있습니다.

## 복소수 연산

HP 35s에서는 다음과 같은 형태의 복소수를 사용할 수 있습니다.

$$x+iy \quad x+yi \quad r\theta a$$

복소수 산술(+, -, ×, ÷), 복소수 삼각법(사인, 코사인, 탄젠트) 및 수학 함수  $-z$ ,  $1/z$ ,  $z_1^{z_2}$ ,  $\ln z$  및  $e^z$ 에 대한 연산이 있습니다. 여기서  $z_1$  및  $z_2$ 는 복소수입니다.

$x+yi$  형태는 ALG 모드에서만 사용할 수 있습니다.

### 복소수를 입력하려면

형태:  $x+iy$

1. 실수부를 입력합니다.
2. **i**를 누릅니다.
3. 허수 부를 입력합니다.

형태:  $x+yi$

1. 실수부를 입력합니다.
2. **+**를 누릅니다.
3. 허수 부를 입력합니다.
4. **i**를 누릅니다.

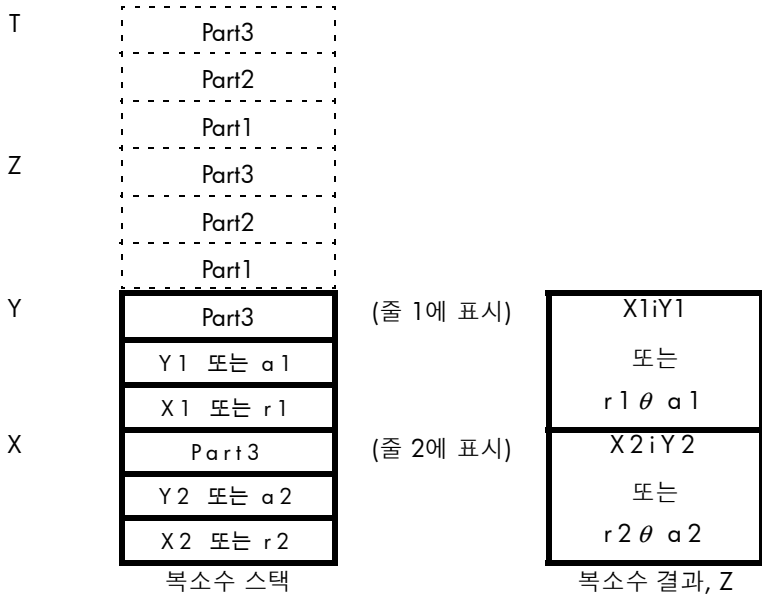
형태:  $r\theta a$

1.  $r$ 의 값을 입력합니다.
2. **→** **θ**를 누릅니다.
3.  $\theta$ 의 값을 입력합니다.

이 장의 예에서는 달리 명시되지 않은 경우 모두 RPN 모드를 사용합니다.

## ✓ 복소수 스택

복소수는 스택 수준의 부분 1과 부분 2를 사용합니다. RPN 모드에서 X 레지스터의 부분 1과 부분 2를 사용하는 복소수는 줄 2에 표시되고, Y 레지스터의 부분 1과 부분 2를 사용하는 복소수는 줄 1에 표시됩니다.












## 복소수 연산

ALG 및 RPN 모드에서 실수 연산과 마찬가지로 복소수 연산을 사용합니다.

### ✓ 하나의 복소수로 연산을 수행하려면

1. 앞에서 설명한 대로 복소수 z를 입력합니다.
2. 복소수 함수를 선택합니다.






### 단일 복소수 $z$ 에 대한 함수

계산:	키:
부호 변경, $-z$	
역수, $1/z$	
자연 로그, $\ln z$	
자연 역대수, $e^z$	
사인 $z$	
코사인 $z$	
탄젠트 $z$	
절대값, $ABS(z)$	
인수 값, $ARG(z)$	

### ✓ 두 개의 복소수로 산술 연산을 수행하려면

1. 앞에서 설명한 대로 첫 번째 복소수  $z_1$ 를 입력합니다.
2. 앞에서 설명한 대로 두 번째 복소수  $z_2$ 를 입력합니다.
3. 산술 연산을 선택합니다.

### 두 개의 복소수 $z_1$ 과 $z_2$ 를 사용한 산술

계산:	키:
더하기, $z_1 + z_2$	
빼기, $z_1 - z_2$	
곱하기, $z_1 \times z_2$	
나누기, $z_1 \div z_2$	
거듭제곱 함수, $z_1^{z_2}$	

**예:**

복소수를 사용한 삼각법 및 산술의 몇 가지 예는 다음과 같습니다.

sin (2i3)을 계산합니다.

**키:**

**DISPLAY** **9** ( $9 \times i \cdot \psi$ )

**2** **i** **3** **SIN**

**디스플레이:**

9.1545*i*-4.1689

**설명:**

디스플레이 형식을 설정합니다.

결과는 9.1545 *i*-4.1689입니다.

다음 수식을 계산합니다.

$$z_1 \div (z_2 + z_3),$$

여기서  $z_1 = 23 i 13$ ,  $z_2 = -2i1$   $z_3 = 4 i - 3$

다음과 같이 계산을 수행합니다.

**키:**

**DISPLAY** **9** ( $9 \times i \cdot \psi$ )

**2** **3** **i** **1** **3** **ENTER**

**2** **+/-** **i** **1** **ENTER**

**4** **i** **3** **+/-** **+**

**÷**

**디스플레이:**

23.0000*i*13.0000

23.0000*i*13.0000

-2.0000*i*1.0000

-2.0000*i*1.0000

23.0000*i*13.0000

2.0000*i*-2.0000

2.5000*i*9000

**설명:**

디스플레이 형식을 설정합니다.

$z_1$ 을 입력합니다.

$z_2$ 를 입력합니다.

$(z_2 + z_3)$ . 결과는 2 *i*-2입니다.

$z_1 \div (z_2 + z_3)$ . 결과는 2.5 *i*9입니다.



$(4i - 2/5) \times (3i - 2/3)$ 를 계산합니다.

**키:**

$\left[ \leftarrow \right]$   $\left[ \text{DISPLAY} \right]$   $\left[ 9 \right]$   $(9 \times i \cdot y)$

$\left[ 4 \right]$   $\left[ i \right]$   $\left[ \cdot \right]$   $\left[ 2 \right]$   $\left[ \cdot \right]$   $\left[ 5 \right]$   $\left[ +\frac{\_}{\_} \right]$

$\left[ \text{ENTER} \right]$

$\left[ 3 \right]$   $\left[ i \right]$   $\left[ \cdot \right]$   $\left[ 2 \right]$   $\left[ \cdot \right]$   $\left[ 3 \right]$   $\left[ +\frac{\_}{\_} \right]$

$\left[ \times \right]$

**디스플레이:**

4.0000i-0.4000

4.0000i-0.4000

4.0000i-0.4000

3i-0.2/3

11.7333i-3.8667

**설명:**

디스플레이 형식을 설정합니다.

4i-2/5를 입력합니다.

3i-2/3을 입력합니다.

결과는 11.7333i-3.8667입니다.

$e^{z-2}$ 을 계산합니다. 여기서  $z = (1/i)$ 입니다.

**키:**

$\left[ 1 \right]$   $\left[ i \right]$   $\left[ 1 \right]$   $\left[ \text{ENTER} \right]$

$\left[ 2 \right]$   $\left[ +\frac{\_}{\_} \right]$   $\left[ y^x \right]$

$\left[ \leftarrow \right]$   $\left[ e^x \right]$

**디스플레이:**

1.0000i-1.0000

1.0000i-1.0000

0.0000i-5.0000

0.8776i-0.4794

**설명:**

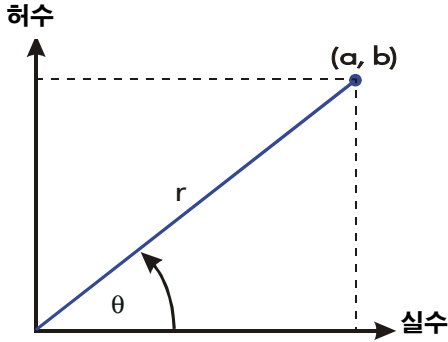
1i를 입력합니다.

Z-2의 중간 결과입니다.  
결과는 0i-5입니다.

최종 결과는  
0.8776i-0.4794입니다.

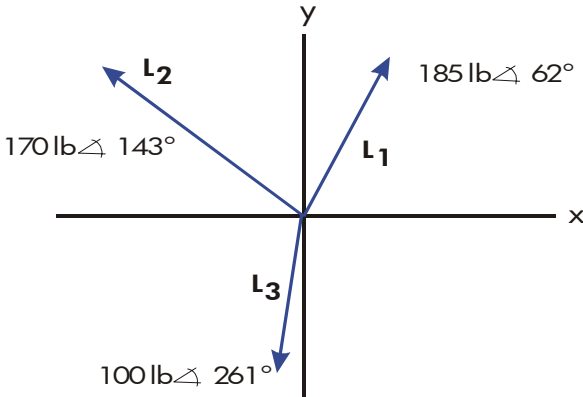
## 극좌표 표기법으로 복소수 사용

많은 응용 프로그램은 극좌표 형태나 극좌표 표기법에 복소수를 사용합니다. 이러한 형태에서는 복소수와 마찬가지로 숫자 쌍을 사용하므로 복소수 연산을 사용하여 해당 숫자로 연산을 수행할 수 있습니다.



## 예: 벡터 더하기

다음 세 개의 로드를 더합니다.



**키:**  
 [MODE] [1] (10 DEG)  
 [←] [DISPLAY] [·] [0]  
 (10rBa)  
 [1] [8] [5] [↵] [0]  
 [6] [2] [ENTER]  
 [1] [7] [0] [↵] [0]  
 [1] [4] [3] [ENTER]  
 [1] [0] [0] [↵] [0]  
 [2] [6] [1] [+]  
 [+]  
 [↵] [>]

**디스플레이:**

185.0000062.0000  
 185.0000062.0000  
 170.00000143.00...  
 170.00000143.000→  
 185.0000062.0000  
 151.45290178.660→  
 178.93720111.148→  
 ←9

**설명:**

도 모드를 설정합니다.  
 복소수 모드를 설정합니다.  
 L<sub>1</sub>을 입력합니다.  
 L<sub>2</sub>를 입력합니다.  
 L<sub>3</sub>을 입력하고 L<sub>2</sub> + L<sub>3</sub>을 더합니다.  
 L<sub>1</sub> + L<sub>2</sub> + L<sub>3</sub>을 더합니다.  
 화면을 스크롤하여 결과의 나머지 부분을 확인합니다.

복소수 형태가 서로 다른 숫자로 복소수 연산을 수행할 수 있습니다. 그러나 결과 형태는 [DISPLAY] 메뉴의 설정에 따라 달라집니다.

$1i + 3\theta 10 + 5\theta 30$ 을 계산합니다.

**키:**  
[MODE] [1] (1 DEG)

[ $\leftarrow$ ] [DISPLAY] [.] [0]

(1  $\theta$  r  $\theta$  a)

[1] [i] [1] [ENTER]

1.4142045.0000

1.4142045.0000

[3] [ $\leftarrow$ ] [0] [1] [0]

3.0000010.0000

[ENTER]

3.0000010.0000

[5] [ $\leftarrow$ ] [0] [3] [0]

1.4142045.0000

[+]

7.8861022.5241

[+]

9.2088025.8898

**디스플레이:**

**설명:**

도 모드를 설정합니다.

복소수 모드를 설정합니다.

1i을 입력합니다.

$3\theta 10$ 을 입력합니다.

$5\theta 30$ 을 입력하고  $3\theta 10$ 을 더합니다.

1i을 더합니다. 결과는  $9.2088\theta 25.8898$ 입니다.

---

## 방정식의 복소수

방정식에 복소수를 입력할 수 있습니다. 방정식을 표시할 때 모든 숫자 형태는 입력된 대로 표시됩니다(예:  $xiy$  또는  $r\theta a$ ).

방정식을 계산하고 변수 값을 묻는 프롬프트가 표시될 때 복소수를 입력할 수 있습니다. 결과 값과 형태는 디스플레이 설정에 의해 제어됩니다. 이것은 **ALG** 모드의 계산과 동일합니다.

복소수가 포함된 방정식을 풀고 적분할 수 있습니다.

---

## 프로그램의 복소수

프로그램에서 복소수를 입력할 수 있습니다. 예를 들어,  $1i2+3010+5$ 와 같이 입력합니다.

프로그램의 0 30은 다음과 같습니다.

**프로그램 줄: (ALG 모드)**  
F001 LBL F  
F002 1i2+3010+5030  
F003 RTN

**설명**  
프로그램을 시작합니다.

프로그램을 실행 중이며 INPUT 명령에서 값을 묻는 프롬프트를 표시할 때 복소수를 입력할 수 있습니다. 결과 값과 형태는 디스플레이 설정에 의해 제어됩니다.

복소수가 포함된 프로그램의 값을 구하고 적분할 수도 있습니다.









## 산술 벡터

---

수학적 관점에서 벡터는 한 행이나 한 열에 정렬된 2개 이상의 원소 배열입니다.

두세 개의 구성 요소가 있고 위치, 속도, 가속도, 힘, 모멘트, 선형 및 각운동량, 각 속도 및 가속도 등의 물리량을 나타내는 데 사용할 수 있는 물리적 벡터입니다.

### 벡터를 입력하려면

1.   를 누릅니다.
2. 벡터의 첫 번째 숫자를 입력합니다.
3.  ,  를 누르고 2D 또는 3D 벡터의 두 번째 숫자를 입력합니다.
4.  ,  를 누르고 3D 벡터의 세 번째 숫자를 입력합니다.



HP 35s는 3차원이 넘는 벡터를 처리할 수 없습니다.

---

## 벡터 연산

### ✓ 덧셈 및 뺄셈:

벡터를 더하고 빼려면 두 벡터 피연산자의 길이가 같아야 합니다. 길이가 서로 다른 벡터를 더하거나 빼려고 하면 "INVALID DATA"라는 오류 메시지가 표시됩니다.

1. 첫 번째 벡터를 입력합니다.
2. 두 번째 벡터를 입력합니다.
3.  또는  를 누릅니다.

✓ [1.5,-2.2]+[-1.5,2.2] 계산

키:	디스플레이:	설명:
<b>MODE</b> <b>5</b> (5RPN)		RPN 모드로 전환합니다 (필요한 경우).
<b>→</b> <b> </b> <b>1</b> <b>.</b> <b>5</b> <b>↵</b>	[1.5000,-2.2000]	[1.5,-2.2]를 입력합니다.
<b>,</b> <b>+/-</b> <b>2</b> <b>.</b> <b>2</b>	[1.5000,-2.2000]	
<b>ENTER</b>		
<b>→</b> <b> </b> <b>+/-</b> <b>1</b> <b>.</b> <b>5</b>	[1.5000,-2.2000]	[-1.5,2.2]를 입력합니다.
<b>↵</b> <b>,</b> <b>2</b> <b>.</b> <b>2</b>	[-1.5,2.2]	
<b>+</b>	0.0000	두 벡터를 더합니다.
	[0.0000,0.0000]	

[ -3.4,4.5 ] - [ 2.3,1.4 ] 계산

키:	디스플레이:	설명:
<b>MODE</b> <b>4</b> (4ALG)		ALG 모드로 전환합니다.
<b>→</b> <b> </b> <b>+/-</b> <b>3</b> <b>.</b> <b>4</b>	[ -3.4,4.5 ]	[ -3.4,4.5 ]를 입력합니다.
<b>↵</b> <b>,</b> <b>4</b> <b>.</b> <b>5</b> <b>&gt;</b>		
<b>-</b> <b>→</b> <b> </b> <b>2</b> <b>.</b> <b>3</b>	← [ -3.4,4.5 ] - [ 2.3,1.4 ]	[ 2.3,1.4 ]를 입력합니다.
<b>↵</b> <b>,</b> <b>1</b> <b>.</b> <b>4</b>		
<b>ENTER</b>	[ -3.4,4.5 ] - [ 2.3,1.4 ]	두 벡터를 뺍니다.
	[ -5.7000,3.1000 ]	

✓ 스칼라 곱셈 및 나눗셈:

1. 벡터를 입력합니다.
2. 스칼라를 입력합니다.
3. 곱셈의 경우 **x**를 누르고, 나눗셈의 경우 **÷**를 누릅니다.



✓ [3,4]x5 계산

키:	디스플레이:	설명:
<b>MODE</b> <b>5</b> (5RPN)		RPN 모드로 전환합니다.
<b>→</b> <b>[ ]</b> <b>3</b> <b>←</b> <b>,</b> <b>4</b>	[3.0000,4.0000]	[3,4]를 입력합니다.
<b>ENTER</b>	[3.0000,4.0000]	
<b>5</b>	[3.0000,4.0000] 5_	스칼라로 5를 입력합니다.
<b>x</b>	0.0000 [15.0000,20.0000]	곱셈을 수행합니다.

[2,4]÷2 계산

키:	디스플레이:	설명:
<b>MODE</b> <b>4</b> (4ALG)		ALG 모드로 전환합니다.
<b>→</b> <b>[ ]</b> <b>±/√</b> <b>2</b> <b>←</b>	[-2,4]_	[2,4]를 입력합니다.
<b>,</b> <b>4</b> <b>→</b>		
<b>÷</b> <b>2</b>	[-2,4]÷2	스칼라로 5를 입력합니다.
<b>ENTER</b>	[-2,4]÷2 [-1.0000,2.0000]	나눗셈을 수행합니다.

## 벡터의 절대값

절대값 함수 "ABS"를 벡터에 적용하면 벡터의 크기가 구해집니다. 벡터  $A=(A_1, A_2, \dots, A_n)$ 의 경우 크기는  $|A| = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$  으로 정의됩니다.

1. **→** **ABS** 를 누릅니다.
2. 벡터를 입력합니다.
3. **ENTER** 를 누릅니다.

예를 들어, 벡터 [5,12]의 절대값은 다음과 같습니다.

**→** **ABS** **→** **[ ]** **5** **←** **,** **12** **ENTER**. 결과는 13입니다. RPN 모드에서는 **MODE** **5** (5RPN) **→** **[ ]** **5** **←** **,** **12** **→** **ABS** 입니다.

# 내적

DOT 함수는 길이가 같은 두 벡터의 내적을 계산하는 데 사용됩니다. 길이가 서로 다른 두 벡터의 내적을 계산하려고 하면 "INVALID DATA"라는 오류 메시지가 표시됩니다.

2D 벡터 [A, B], [C, D]의 경우 내적은  $[A, B] \cdot [C, D] = A \times C + B \times D$ 로 정의됩니다.

3D 벡터 [A, B, X], [C, D, Y]의 경우 내적은  $[A, B, X] \cdot [C, D, Y] = A \times C + B \times D + X \times Y$ 로 정의됩니다.

1. 첫 번째 벡터를 입력합니다.
2. **[X]**를 누릅니다.
3. 두 번째 벡터를 입력합니다.
4. **[ENTER]**를 누릅니다.

참고: 여기서 **[X]** 부호는 "외적"이 아니라 "내적"을 의미합니다. 외적의 경우 17장을 참조하십시오.

두 벡터 [1,2]와 [3,4]의 내적 계산

키:	디스플레이:	설명:
<b>[MODE]</b> <b>[4]</b> (4RLG)		ALG 모드로 전환합니다.
<b>[&gt;]</b> <b>[1]</b> <b>[,]</b> <b>[2]</b>	[1,2]_	첫 번째 벡터 [1,2]를 입력합니다.
<b>[&gt;]</b>		
<b>[X]</b> <b>[&gt;]</b> <b>[3]</b> <b>[,]</b> <b>[4]</b>	[1,2]×[3,4]	내적에 대해 <b>[X]</b> 를 실행하고 두 번째 벡터를 입력합니다.
<b>[4]</b>		두 벡터의 내적은 11입니다.
<b>[ENTER]</b>	11.0000	

✓ 두 벡터 [9,5]와 [2,2]의 내적 계산

키:	디스플레이:	설명:
<b>[MODE]</b> <b>[5]</b> (5RPN)		RPN 모드로 전환합니다.
<b>[&gt;]</b> <b>[9]</b> <b>[,]</b> <b>[5]</b>	[9.0000,5.0000]	첫 번째 벡터 [9,5]를 입력합니다.
<b>[ENTER]</b>	[9.0000,5.0000]	
<b>[&gt;]</b> <b>[2]</b> <b>[,]</b> <b>[2]</b>	[9.0000,5.0000] [2,2]	두 번째 벡터 [2,2]를 입력합니다.
<b>[MODE]</b> <b>[5]</b> (5RPN)	28.0000	내적에 대해 <b>[X]</b> 를 누르면 두 벡터의 내적은 28입니다.

## 10-4 산술 벡터

## 벡터 사이의 각도

두 벡터 A와 B 사이의 각도는  $\theta =$

$\text{ACOS}(A \cdot B / |A||B|)$ 로 구할 수 있습니다.

두 벡터  $A=[1,0], B=[0,1]$  사이의 각도 구하기

**키:**

**MODE** **4** (4ALG)

**MODE** **1** (1DEG)

**ACOS**

**1** **0** **,** **0**

**>**

**x** **ACOS** **1** **0** **,** **0** **>**

**1** **>**

**÷** **ACOS** **ABS** **1** **0**

**1** **,** **0** **>**

**÷** **ACOS** **ABS** **0** **1**

**0** **,** **1**

**ENTER**

**디스플레이:**

ACOS()

ACOS([1,0])

ACOS([1,0]×[0,1])

←,1]÷ABS([1,0])→

←1,0]÷ABS([0,1])→

ACOS([1,0]×[0,1])

90.0000

**설명:**

ALG 모드로 전환합니다.

도(Degree) 모드를 설정합니다.

아코사인 함수입니다.

벡터 A [1,0]을 입력합니다.

A와 B의 내적에 대한 벡터 B [0,1]을 입력합니다.

벡터 A [1,0]의 크기를 구합니다.

벡터 B [0,1]의 크기를 구합니다.

두 벡터 사이의 각도는 90입니다.

✓ 두 벡터  $A=[3,4], B=[0,5]$  사이의 각도 구하기

**키:**

**MODE** **5** (5RPN)

**MODE** **1** (1DEG)

**1** **3** **,** **4**

**ENTER** **0** **,** **5**

**,** **5** **x**

**1** **3** **,** **4**

**ABS** 5.0000

**1** **0** **,** **5** 5.0000

**ABS** 5.0000

**x** 20.0000

25.0000

**디스플레이:**

**설명:**

RPN 모드로 전환합니다.

도 모드를 설정합니다.

두 벡터의 내적을 구합니다.

벡터 [3,4]의 크기를 구합니다.

벡터 [0,5]의 크기를 구합니다.

두 벡터를 곱합니다.



90  
0.8000  
90  
36.8699

두 값을 나눕니다.  
두 벡터 사이의 각도는  
36.8699입니다.



## 방정식의 벡터

벡터를 마치 실수처럼 방정식과 방정식 변수에 사용할 수 있습니다. 변수를 묻는 프롬프트가 표시될 때 벡터를 입력하면 됩니다.

벡터가 포함된 방정식의 값을 구할 수는 있지만 미지수가 벡터인 경우에는 값을 구하기 어렵습니다.

벡터가 포함된 방정식을 적분할 수는 있지만 방정식 결과가 실수이거나 1D 벡터 또는 2차 및 3차 원소가 0인 벡터여야 합니다.

## 프로그램의 벡터

벡터를 실수나 복소수와 같은 방식으로 프로그램에 사용할 수 있습니다.

예를 들어, 프로그램의  $[5, 6] + 2 \times [7, 8] \times [9, 10]$ 은 다음과 같습니다.

### 프로그램 줄:

```
G0001 LBL G
G0002 [5,6] + 2 x [7,8] x [9,10]
G0003 RTN
```

### 설명:

프로그램을 시작합니다.  
[5,6]

변수 값을 묻는 프롬프트가 표시될 때 벡터를 입력하면 됩니다. 벡터가 포함된 프로그램을 값을 구하거나 적분하는 데 사용할 수 있습니다.

---

## 변수 또는 레지스터에서 벡터 만들기

메모리 변수의 내용, 스택 레지스터 또는 간접 레지스터의 값이 포함된 벡터를 실행 모드나 프로그램 모드에서 만들 수 있습니다.

ALG 모드에서는 **[>]** **[I]**를 눌러 벡터 입력을 시작합니다. RPN 모드는 **[EQN]** 키를 먼저 누른 다음 **[>]** **[I]**를 눌러야 한다는 점을 제외하고 ALG 모드와 유사하게 작동합니다.

문자 변수에 저장된 값이 있는 원소를 입력하려면 **[RCL]**과 변수 글자를 누릅니다.


스택 레지스터에서 원소를 입력하려면 **[R↓]** 키를 누르고 **[>]** 또는 **[<]** 키를 사용하여 밑줄 기호가 스택 레지스터 아래에 오도록 이동하고 **[ENTER]**를 누릅니다.


I 또는 J 레지스터의 값이 간접적으로 나타내는 원소를 입력하려면 **[RCL]**과 (I) 또는 (J)를 누릅니다.

예를 들어, RPN 모드에서 [ C, REGZ, (I) ] 벡터를 구성하려면 **[EQN]** **[>]** **[I]**를 누른 다음 **[RCL]** **[C]** **[←]** **[,]** **[R↓]** **[>]** **[ENTER]** **[←]** **[,]** **[RCL]** **[(I)]** **[ENTER]**를 누릅니다.







## 진법 변환과 산술 및 논리

BASE 메뉴( **BASE**)를 사용하면 숫자를 입력하고 숫자 표시를 10진수, 2진수, 8진수, 16진수 등으로 적용할 수 있습니다.

LOGIC 메뉴( **LOGIC**)를 통해 논리 함수에 액세스할 수 있습니다.

### BASE 메뉴

메뉴 레이블	설명
DEC	10진수 모드입니다. 일반 계산기 모드입니다.
HEX	16진수 모드입니다. 이 모드가 활성화 상태이면 <b>HEX</b> 표시기가 나타납니다. 숫자가 16진수 형식으로 표시됩니다. RPN 모드에서 <b>SIN</b> , <b>COS</b> , <b>TAN</b> , $\sqrt{x}$ , $y^x$ 및 $1/x$ 키는 숫자 A-F를 입력하는 바로 가기로 사용됩니다. ALG 모드에서는 <b>RCL</b> A, B, C, D, E 또는 F를 눌러 숫자 A-F를 입력합니다.
OCT	8진수 모드입니다. 이 모드가 활성화 상태이면 <b>OCT</b> 표시기가 나타납니다. 숫자가 8진수 형식으로 표시됩니다.
BIN	2진수 모드입니다. 이 모드가 활성화 상태이면 <b>BIN</b> 표시기가 나타납니다. 숫자가 2진수 형식으로 표시됩니다. 숫자가 12자리수보다 큰 경우   및   키를 사용하여 전체 숫자를 볼 수 있습니다. 이 장의 뒷부분에 있는 “긴 2진수의 구간”을 참조하십시오.
d	숫자 끝에 표시하여 이 숫자가 10진수임을 나타냅니다.
h	숫자 끝에 표시하여 이 숫자가 16진수임을 나타냅니다. 16진수를 입력하려면 숫자 뒤에 “h”를 입력합니다.

o	숫자 끝에 표시하여 이 숫자가 8진수임을 나타냅니다. 8진수를 입력하려면 숫자 뒤에 "o"를 입력합니다.
b	숫자 끝에 표시하여 이 숫자가 2진수임을 나타냅니다. 2진수를 입력하려면 숫자 뒤에 "b"를 입력합니다.

## 예: 숫자의 진법 변환

다음 키 입력은 다양한 진법 변환을 수행합니다.

125.99<sub>10</sub>을 16진수, 8진수 및 2진수로 변환합니다.

키:	디스플레이:	설명:
	70h	10진수를 16진수로 변환합니다.
(2HEX)		다.
(3OCT)	175o	8진수로 변환합니다.
(4BIN)	1111101b	2진수로 변환합니다.
(1DEC)	125.0000	

참고: 10진수를 사용하지 않는 경우에는 디스플레이에 숫자의 정수부만 표시됩니다. 그러나 지우기 기능을 수행하지 않았다면 소수부가 기억되며, 10진수를 선택하면 소수부가 표시됩니다.

24FF<sub>16</sub>을 2진수로 변환합니다. 2진수는 길이가 14자리(디스플레이 최대 자릿수)를 초과합니다.

키:	디스플레이:	설명:
(2HEX)	24FFh	키를 사용하여 "F"를 입력합니다.
(6h)		
(4BIN)	1001001111111111	2진수 전체가 다 들어가지 않습니다.  표시기는 숫자가 오른쪽으로 계속됨을 나타냅니다.

## 11-2 진법 변환과 산술 및 논리





숫자의 나머지 부분을 표시합니다. 전체 숫자는 10010011111111<sub>b</sub>입니다.



10010011111111

처음 14자릿수를 다시 표시합니다.



9,471.0000

10진수로 복원합니다.

**[BASE]** 메뉴를 사용하면 피연산자 뒤에 n 진수 기호 b/o/d/h를 입력하여 다른 진법 모드에서 2/8/10/16진수로 나타낼 수 있습니다. 진수 기호가 없는 숫자는 10진수입니다.

참고:

ALG 모드에서는 다음과 같이 동작합니다.

1. 결과의 진법 모드는 현재 진법 모드 설정에 의해 결정됩니다.
2. 활성 명령줄이 없는 경우(줄 1에 깜박이는 커서가 없음) 진수를 변경하면 줄 2가 새로운 진수로 업데이트됩니다.
3. **[ENTER]**를 누르거나 진법 모드를 변경하면 계산기가 자동으로 현재 진수 기호 b/o/h를 결과 뒤에 표시하여 줄 2의 숫자가 2/8/16진수임을 나타냅니다.
4. 수식을 다시 편집하려면 **[<]** 또는 **[>]**를 누릅니다.

RPN 모드에서는 다음과 같이 동작합니다.

줄 2에 숫자를 입력하고 **[ENTER]**를 누른 다음 진법 모드를 변경하면 계산기에서 줄 1과 줄 2의 숫자 진수를 변환하고 숫자 뒤에 기호 b/o/h가 추가되어 2/8/16진수를 나타냅니다.

줄 2에서 다음 화면의 내용을 보려면 **[>]** 또는 **[<]**를 눌러 화면을 변경합니다.

## LOGIC 메뉴

메뉴 레이블	설명
AND	두 인수의 논리적 비트 단위 "AND"입니다. 예: AND(1100b,1010b)=1000b
XOR	두 인수의 논리적 비트 단위 "XOR"입니다. 예: XOR(1101b,1011b)=110b
OR	두 인수의 논리적 비트 단위 "OR"입니다. 예: OR(1100b,1010b)=1110b
NOT	인수의 1의 보수를 반환합니다. 결과의 각 비트는 인수에 있는 해당 비트의 보수입니다. 예: NOT(1011b)= 1111111111111111111111111111111110100b
NAND	두 인수의 논리적 비트 단위 "NAND"입니다. 예: NAND(1100b,1010b)=11111111111111111111 1111111110111b
NOR	두 인수의 논리적 비트 단위 "NOR"입니다. 예: NOR(1100b,1010b)= 1111111111111111111111111111111110001b

"AND", "OR", "XOR", "NOT", "NAND", "NOR"을 논리 함수로 사용할 수 있습니다. 분수, 복소수, 벡터 인수는 논리 함수에서 "INVALID DATA"로 표시됩니다.

## 2, 8 및 16진수의 산술

모든 진수에서  $\boxed{+}$ ,  $\boxed{-}$ ,  $\boxed{\times}$  및  $\boxed{\div}$ 를 사용하여 산술 연산을 수행할 수 있습니다. HEX 모드에서 실제로 비활성화되는 기능 키는  $\boxed{\sqrt{x}}$ ,  $\boxed{e^x}$ ,  $\boxed{\ln}$ ,  $\boxed{y^x}$ ,  $\boxed{1/x}$  및  $\boxed{\Sigma+}$  뿐입니다. 그러나 숫자의 소수부가 잘리기 때문에 산술 이외의 연산은 대부분 의미 있는 결과를 제공하지 않습니다.

2, 8 및 16진수의 산술은 2의 보수 형태이며 정수만 사용합니다.

- 숫자에 소수부가 있는 경우 정수부만 산술 계산에 사용됩니다.

### 11-4 진법 변환과 산술 및 논리

■ 연산 결과는 항상 정수입니다(모든 소수부가 잘림).

변환 시에는 숫자 표시만 변경되고 X 레지스터의 실제 숫자는 변경되지 않지만 산술은 X 레지스터의 숫자를 변경합니다.

연산 결과를 유효한 비트로 나타낼 수 없는 경우 디스플레이에 **OVERFLOW**가 표시된 다음 가능한 가장 큰 양수 또는 음수가 표시됩니다.

예:

16진수, 8진수 및 2진수 모드의 산술 예는 다음과 같습니다.

$$12F_{16} + E9A_{16} = ?$$

키:

**[2nd]** **[BASE]** **[2]** (2HEX)

✓ **[1]** **[2]** **[1/x]** **[2nd]** **[BASE]** **[6]**  
(6h) **[ENTER]** **[y<sup>x</sup>]** **[9]** **[SIN]** **[>]**  
**[BASE]** **[6]** (6h) **[+]**

디스플레이:

FC9h

설명:

16진수를 설정하고 **HEX** 표시기가 켜집니다. 결과입니다.

$$7760_8 - 4326_8 = ?$$

**[2nd]** **[BASE]** **[3]** (3OCT)

✓ **[7]** **[7]** **[6]** **[0]** **[2nd]** **[BASE]**  
**[7]** (7o) **[ENTER]** **[4]** **[3]** **[2]**  
**[6]** **[2nd]** **[BASE]** **[7]** (7o) **[=]**

7711o

3432o

8진수를 설정하고 **OCT** 표시기가 켜집니다. 표시된 숫자를 8진수로 변환합니다. 결과입니다.

$$100_8 \div 5_8 = ?$$

✓ **[1]** **[0]** **[0]** **[2nd]** **[BASE]** **[7]**  
(7o) **[ENTER]** **[5]** **[2nd]** **[BASE]**  
**[7]** (7o) **[÷]**

14o

결과의 정수부입니다.

$$5A0_{16} + 1001100_2 = ?$$

✓ **[2nd]** **[BASE]** **[2]** (2HEX) **[5]**  
**[SIN]** **[0]** **[2nd]** **[BASE]** **[6]**  
(6h) **[ENTER]**

5A0h

16진수를 설정하고 **HEX** 표시기가 켜집니다.



- ▶ BASE 4 (4BIN)     1111111111111111▶     2진수 버전이며, ▶는 나머지 숫자를 나타냅니다. 최상위 비트가 1이기 때문에 이 숫자는 음수입니다.
  
- ▶ >     ◀111111111111101▶     한 화면을 스크롤하여 숫자의 나머지 부분을 표시합니다.
  
- ▶ >     ◀11011110b     맨 오른쪽 구간을 표시합니다.
  
- ▶ BASE 1 (1DEC)     -546.0000     음수 10진수입니다.

## 숫자 범위

36비트 2진수 크기로 16진수(9자릿수), 8진수(12자릿수) 및 2진수(36자릿수)로 나타낼 수 있는 숫자 범위와 이러한 다른 진법으로 변환할 수 있는 10진수(11자릿수) 범위가 결정됩니다.

### 진법 변환의 숫자 범위

진수	최대 크기의 양의 정수	최대 크기의 음의 정수
16진수	7FFFFFFFFh	800000000h
8진수	37777777777o	400000000000o
2진수	01111111111111111111111111111111 1111111111111111b	10000000000000000000000000000000 00000000000000b
10진수	34,359,738,367	-34,359,738,368

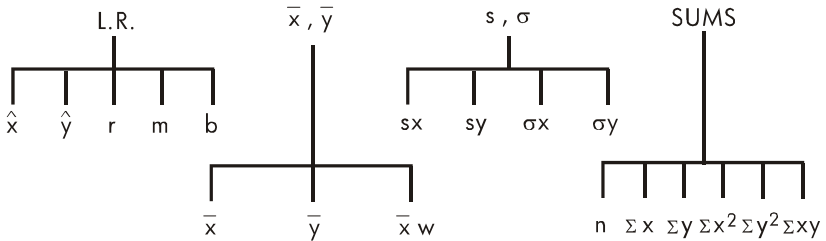
10 이외의 진수를 선택한 경우 이 범위를 벗어난 숫자를 입력할 수 없습니다.



## 통계 연산

HP 35s의 통계 메뉴에서는 1개 또는 2개 변수 데이터(실수) 세트를 통계적으로 분석하는 기능을 사용할 수 있습니다.

- 평균, 표본 표준 편차 및 모집단 표준 편차
- 선형 회귀 및 선형 추산( $\hat{x}$  및  $\hat{y}$ )
- 가중 평균( $\gamma$ 가중치가 적용된  $x$ )
- 합계 통계:  $n$ ,  $\Sigma x$ ,  $\Sigma y$ ,  $\Sigma x^2$ ,  $\Sigma y^2$  및  $\Sigma xy$



## 통계 데이터 입력

1개 및 2개 변수 통계 데이터는  $\Sigma+$  (또는  $\leftarrow \Sigma-$ ) 키를 사용하여 유사한 방식으로 입력(또는 삭제)됩니다. 데이터 값은 이름이 SUMS 메뉴에 표시되는 6개 통계 레지스터(-27 ~ -32)에 합계 통계로 누적됩니다.  $\Sigma$  (SUMS)를 누르고  $n \Sigma x \Sigma y \Sigma x^2 \Sigma y^2 \Sigma xy$ 를 확인합니다.

### 참고



새 통계 데이터 세트를 입력하기 전에 항상 통계 레지스터를 지웁니다( $\leftarrow$  CLEAR 4) (4번) 누름).

## 1개 변수 데이터 입력

1. **[ $\rightarrow$ ] [CLEAR] [4] (4 $\Sigma$ )**를 눌러 기존 통계 데이터를 지웁니다.
2. 각  $x$  값을 입력하고 **[ $\Sigma+$ ]**를 누릅니다.
3. 디스플레이에 현재 누적된 통계 데이터 값의 개수인  $n$ 이 표시됩니다.

**[ $\Sigma+$ ]**를 누르면 이미 Y 레지스터에 있는 값이  $y$  값으로 누적되기 때문에 실제로 2개의 변수가 통계 레지스터에 입력됩니다. 이런 이유로 계산기는 선형 회귀를 수행하고,  $x$  데이터만 입력한 경우 또는 동일하지 않은 개수의  $x$  값과  $y$  값을 입력한 경우에도  $y$ 를 기반으로 값을 표시합니다. 오류가 발생하지는 않지만 결과에 명확한 의미가 없습니다.

값을 입력한 직후에 디스플레이로 다시 호출하려면 **[ $\rightarrow$ ] [LAST $x$ ]**를 누릅니다.

## ✓ 2개 변수 데이터 입력

데이터가 변수 쌍인 경우 먼저 종속 변수(쌍의 두 번째 변수)를 입력하고 **[ENTER]**를 누른 다음 독립 변수(쌍의 첫 번째 변수)를 입력하고 **[ $\Sigma+$ ]**를 누릅니다.

1. **[ $\rightarrow$ ] [CLEAR] [4] (4 $\Sigma$ )**를 눌러 기존 통계 데이터를 지웁니다.
2.  $y$  값을 먼저 입력하고 **[ENTER]**를 누릅니다.
3. 해당  $x$  값을 입력하고 **[ $\Sigma+$ ]**를 누릅니다.
4. 디스플레이에 누적된 통계 데이터 쌍의 개수인  $n$ 이 표시됩니다.
5.  $x, y$  쌍의 입력을 계속합니다.  $n$ 이 각 입력으로 업데이트됩니다.

$x$  값을 입력한 직후에 디스플레이로 다시 호출하려면 **[ $\rightarrow$ ] [LAST $x$ ]**를 누릅니다.

## 데이터 입력의 오류 수정

통계 데이터를 입력할 때 실수를 한 경우 잘못된 데이터를 삭제하고 올바른 데이터를 추가합니다.  $x, y$  쌍의 값이 하나만 잘못된 경우에도 두 값을 모두 삭제하고 다시 입력해야 합니다.



## 통계 데이터를 수정하려면

1.  $\Sigma+$ 를 누르는 대신 잘못된 데이터를 다시 입력하고  $\leftarrow$   $\Sigma-$ 를 누릅니다. 이렇게 하면 값이 삭제되고  $n$ 이 감소합니다.
2.  $\Sigma+$ 를 사용하여 올바른 값을 입력합니다.

✓ 방금 입력한 값이 잘못된 값인 경우  $\leftarrow$   $\text{LAST } x$ 를 눌러 검색하고  $\leftarrow$   $\Sigma-$ 를 눌러 삭제합니다. 잘못된  $y$ 값은 여전히  $Y$  레지스터에 있으며, 해당  $x$  값이  $\text{LAST } X$  레지스터에 저장되었습니다. 잘못된 통계 데이터를 삭제한 후 계산기는 줄 1에  $Y$  레지스터의 값을 표시하고 줄 2에  $n$  값을 표시합니다.

예:

왼쪽의  $x, y$  값을 입력한 다음 오른쪽에 표시된 대로 수정합니다.

초기 $x, y$	수정된 $x, y$
20, 4	20, 5
400, 6	40, 6

키:

디스플레이:

설명:

- |  |   |  |
|--|---|--|
| <p>✓ <math>\rightarrow</math> CLEAR 4 (4<math>\Sigma</math>)</p> <p>✓ 4 ENTER 2 0 <math>\Sigma+</math></p> <p>6 ENTER 4 0 0</p> <p><math>\Sigma+</math></p> <p><math>\rightarrow</math> LAST <math>x</math></p> <p><math>\leftarrow</math> <math>\Sigma-</math></p> <p>✓ 6 ENTER 4 0 <math>\Sigma+</math></p> <p>✓ 4 ENTER 2 0 <math>\leftarrow</math></p> <p><math>\Sigma-</math></p> | <p>4.0000</p> <p>1.0000</p> <p>6.0000</p> <p>2.0000</p> <p>6.0000</p> <p>400.0000</p> <p>6.0000</p> <p>1.0000</p> <p>6.0000</p> <p>2.0000</p> <p>4.0000</p> <p>1.0000</p> | <p>기존 통계 데이터를 지웁니다.</p> <p>첫 번째 새 데이터 쌍을 입력합니다.</p> <p>디스플레이에 입력한 데이터 쌍의 개수인 <math>n</math>이 표시됩니다.</p> <p>마지막 <math>x</math> 값을 다시 호출합니다. 마지막 <math>y</math>는 여전히 <math>Y</math> 레지스터에 있습니다.</p> <p>마지막 데이터 쌍을 삭제합니다.</p> <p>마지막 데이터 쌍을 다시 입력합니다.</p> <p>첫 번째 데이터 쌍을 삭제합니다.</p> |
|--|---|--|



**예: 평균(1개 변수)**

생산 관리자인 May Kitt는 특정 프로세스의 평균 소요 시간을 확인하려고 합니다. 임의로 6명의 직원을 선택하고 각 직원의 프로세스 수행을 관찰한 다음 소요 시간을 분 단위로 기록합니다.

15.5	9.25	10.0
12.5	12.0	8.5

시간의 평균을 계산합니다. 모든 데이터를 x 값으로 처리합니다.

키:	디스플레이:	설명:
CLEAR  4 (4Σ)		통계 레지스터를 지웁니다.
1  5  .  5  Σ+	1.0000	첫 번째 시간을 입력합니다.
9  .  2  5  Σ+  1  0		나머지 데이터를 입력합니다.
Σ+  1  2  .  5  Σ+  1	6.0000	6개의 데이터 요소가 누적됩니다.
2  Σ+  8  .  5  Σ+		니다.
(x̄y)	—	프로세스를 완료하는 평균 시간을 계산합니다.
	$\bar{x}$ $\bar{y}$ $\bar{x}w$	
	11.2917	

**예: 가중 평균(2개 변수)**

한 제조 회사에서는 특정 부품을 1년에 4회 구매합니다. 작년의 구매 내역은 다음과 같았습니다.

부품당 가격(x)	\$4.25	\$4.60	\$4.70	\$4.10
부품 개수(y)	250	800	900	1000

이 부품의 평균 가격(구매 수량의 가중치 적용)을 구합니다. 가격인 x보다 먼저 가중치(빈도)인 y를 입력해야 합니다.

키:	디스플레이:	설명:
CLEAR  4 (4Σ)		통계 레지스터를 지웁니다.
2  5  0  ENTER  4  .		데이터를 입력하고 n을 표시합니다.
2  5  Σ+		
8  0  0  ENTER  4  .		
6  Σ+		
9  0  0  ENTER  4  .		
7  Σ+	900.0000	
	3.0000	

✓ 1 0 0 0 ENTER 4  
 1 Σ+  
 [←] [x̄y] [→] [→] (x̄w)

1,000.0000  
 4.0000  
 x̄ y x̄w  
 4.4314

4개의 데이터 쌍이 누적됩니다.  
 구매 수량의 가중치가 적용된 평균 가격을 계산합니다.

## 표본 표준 편차

표본 표준 편차는 데이터 값이 평균에서 얼마나 분산되어 있는지를 측정하는 값입니다. 표본 표준 편차에서는 데이터가 큰 전체 데이터 세트의 표본이며  $n - 1$ 을 약수로 사용하여 계산된다고 가정합니다.

- $x$  값의 표준 편차를 구하려면 [←] [S.σ] (s<sub>x</sub>)를 누릅니다.
- $y$  값의 표준 편차를 구하려면 [←] [S.σ] [→] (s<sub>y</sub>)를 누릅니다.

이 메뉴의 (s<sub>x</sub>) 및 (s<sub>y</sub>) 항목에 대해서는 다음 절인 “모집단 표준 편차”에서 설명합니다.

### 예: 표본 표준 편차

위의 “평균” 예와 동일한 프로세스 시간을 사용하여, 이제 May Kitt는 프로세스의 표준 편차 시간(s<sub>x</sub>)을 확인하려고 합니다.

15.5	9.25	10.0
12.5	12.0	8.5

시간의 표준 편차를 계산합니다. 모든 데이터를  $x$  값으로 처리합니다.

**키:**  
 [←] [CLEAR] 4 (4Σ)  
 1 5 . 5 Σ+  
 9 . 2 5 Σ+ 1 0  
 Σ+ 1 2 . 5 Σ+ 1  
 2 Σ+ 8 . 5 Σ+  
 [←] [S.σ] (s<sub>x</sub>)










**디스플레이:**  
 1.0000  
 6.0000  
 s<sub>x</sub> s<sub>y</sub> ?x ?y  
 2.5808

**설명:**  
 통계 레지스터를 지웁니다.  
 첫 번째 시간을 입력합니다.  
 나머지 데이터를 입력합니다.  
 6개의 데이터 요소가 입력됩니다.  
 표준 편차 시간을 계산합니다.

## 12-6 통계 연산























## 모집단 표준 편차

모집단 표준 편차는 데이터 값이 평균에서 얼마나 분산되어 있는지를 측정한 값입니다. 모집단 표준 편차에서는 데이터가 *전체* 데이터 세트이며  $n$ 을 약수로 사용하여 계산된다고 가정합니다.

- $x$  값의 모집단 표준 편차를 구하려면     ( $s_x$ )를 누릅니다.
- $y$  값의 모집단 표준 편차를 구하려면      ( $s_y$ )를 누릅니다.

### 예: 모집단 표준 편차

Grandma Hinkle의 장성한 네 아들의 키는 각각 170, 173, 174, 180cm입니다. 키의 모집단 표준 편차를 구해 보십시오.

키:	디스플레이:	설명:
 CLEAR  (4 $\Sigma$ )		통계 레지스터를 지웁니다.
      		데이터를 입력합니다. 4개의 데이터 요소가 누적됩니다.
      	4.0000	
 		
    ( $s_x$ )	$s_x$ $s_y$ $?x$ $?y$ 3.6315	모집단 표준 편차를 계산합니다.

## 선형 회귀

선형 회귀인 L.R.은 *선형 추산*이라고도 하며,  $x, y$  데이터 세트에 가장 맞는 직선을 구하는 통계 방법입니다.

---

### 참고



STAT ERROR 메시지를 방지하려면 L.R. 메뉴의 기능을 실행하기 *전에* 데이터를 입력합니다.

---

## L.R. (선형 회귀) 메뉴

메뉴 키	설명
$\hat{x}$	데이터에 맞게 계산된 선을 기반으로, 제공된 $y$ 의 가상 값에 대한 $x$ 를 추산(추정)합니다.
$\hat{y}$	데이터에 맞게 계산된 선을 기반으로, 제공된 $x$ 의 가상 값에 대한 $y$ 를 추산(추정)합니다.
$r$	$(x, y)$ 데이터의 상관계수입니다. 상관계수는 계산된 선이 데이터에 얼마나 맞는지를 측정하는 $-1$ 에서 $+1$ 사이의 숫자입니다.
$m$	계산된 선의 기울기입니다.
$b$	계산된 선의 $y$ 인터셉트입니다.

- ✓ ■  $x$ (또는  $y$ )에 대한 추정값을 구하려면 제공된  $y$ (또는  $x$ )의 가상 값을 입력하고 **L.R.** ( $\hat{x}$ )(또는 **L.R.** ( $\hat{y}$ )를 누릅니다.
- 데이터에 가장 맞는 선을 정의하는 값을 구하려면 **L.R.** 뒤에  $r$ ,  $m$  또는  $b$ 를 누릅니다.

### 예: 곡선 맞춤

다양한 신종 쌀의 수확량은 질소 비료의 비율에 따라 달라집니다. 다음 데이터의 선형 관계, 즉 상관계수, 기울기 및  $y$  인터셉트를 확인합니다.

<b>X, 질소 적용</b> (헥타르당 kg)	0.00	20.00	40.00	60.00	80.00
<b>Y, 쌀 수확량</b> (헥타르당 미터톤)	4.63	5.78	6.61	7.21	7.78

**키:**

**▶** CLEAR **4** (4 $\Sigma$ )

✓ **4** **.** **6** **3** ENTER **0**  
 $\Sigma+$

✓ **5** **.** **7** **8** ENTER **2**  
**0**  $\Sigma+$

✓ **6** **.** **6** **1** ENTER **4**  
**0**  $\Sigma+$

✓ **7** **.** **2** **1** ENTER **6**  
**0**  $\Sigma+$

✓ **7** **.** **7** **8** ENTER **8**  
**0**  $\Sigma+$

**◀** L.R. **>** **>** (r)

**>**

**>**

**디스플레이:**

7.2100  
4.0000

7.7800  
5.0000

$\hat{x}$   $\hat{y}$  r m b  
0.9880

$\hat{x}$   $\hat{y}$  r m b  
0.0387

$\hat{x}$   $\hat{y}$  r m b  
4.8560

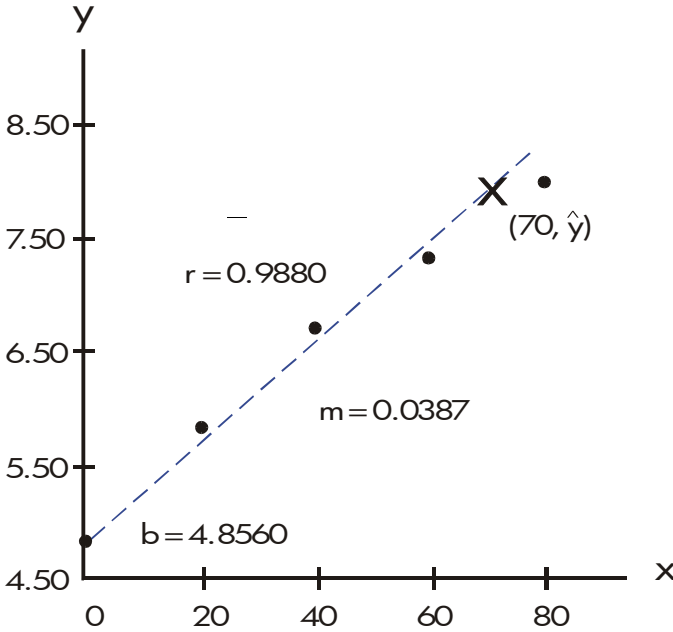
**설명:**

이전 통계 데이터를 모두 지웁니다.  
데이터를 입력하고  $n$ 을 표시합니다.

5개의 데이터 쌍이 입력됩니다.

선형 회귀 메뉴를 표시합니다.  
상관계수이며 데이터가 직선에 가깝게 근접합니다.  
선의 기울기입니다.

$y$  인터셉트입니다.



논에 70kg의 질소 비료를 적용하면 어떻게 되겠습니까? 위의 통계를 기반으로 쌀 수확량을 예측합니다.

키:

✓ [C] [7] [0]

디스플레이:

7.7800  
70\_

설명:

가상 x 값을 입력합니다.

[<] [L.R.] [ > ] ( $\hat{y}$ )

$\hat{x}$   $\hat{y}$  r m b  
7.5615

헥타르당 톤 단위의 예상 수확량입니다.



---

## 데이터 정밀도의 한계



계산기는 유한 정밀도를 사용하므로 반올림으로 인한 계산에 한계가 있습니다. 다음은 두 가지 예입니다.

### 가까운 큰 숫자 정규화

계산기에서 데이터 값의 차이가 비교적 적은 변수의 표준 편차와 선형 회귀를 올바르게 계산하지 못할 수도 있습니다. 이를 방지하려면 각 값을 하나의 중심 값(예: 평균)과의 차이로 입력하여 데이터를 정규화합니다. 정규화된  $x$  값의 경우  $\bar{x}$  및  $\hat{x}$ 의 계산에 이 차이를 다시 더해야 하며  $\hat{y}$  및  $b$ 도 조정해야 합니다. 예를 들어,  $x$  값이 7776999, 7777000 및 7777001이면 데이터를  $-1, 0$  및  $1$ 로 입력한 다음 7777000을  $\bar{x}$  및  $\hat{x}$ 에 다시 더해야 합니다.  $b$ 의 경우  $7777000 \times m$ 을 다시 더해야 합니다.  $\hat{y}$ 를 계산하려면 7777000보다 작은  $x$  값을 제공해야 합니다.

$x$  및  $y$  값의 크기 차이가 매우 큰 경우 유사한 부정확성이 발생할 수 있습니다. 데이터 크기를 조정하면 이 문제를 방지할 수 있습니다.

### 삭제된 데이터의 영향

 를 실행해도 원래 데이터 값에 의해 통계 레지스터에 생성되었을 수 있는 반올림 오류는 삭제되지 않습니다. 잘못된 데이터의 크기가 올바른 데이터에 비해 훨씬 큰 경우가 아니면 이 차이는 심각하지 않습니다. 훨씬 큰 경우에는 모든 데이터를 지우고 다시 입력하는 것이 좋습니다.

# 합계 값 및 통계 레지스터

통계 레지스터는 6개 합계 값의 누적을 저장하는 6개의 고유 메모리 장소입니다.

## 합계 통계

**SUMS** 을 누르면 통계 레지스터의 내용에 액세스할 수 있습니다.

- 누적된 데이터 세트의 개수를 다시 호출하려면 **(n)**을 누릅니다.
- $x$  값의 합계를 다시 호출하려면 **>** ( $\Sigma x$ )를 누릅니다.
- $y$  값의 합계를 다시 호출하려면 **>** **>** ( $\Sigma y$ )를 누릅니다.
- 계산기에서 제공하는 통계 계산뿐 아니라 다른 통계 계산을 수행할 때 사용되는  $x$  및  $y$  값의 곱 합계와 제곱 합계를 다시 호출하려면 **>** **>** **>** ( $\Sigma xy$ ), **>** **>** **>** **>** ( $\Sigma y^2$ ) 및 **>** **>** **>** **>** **>** ( $\Sigma x^2$ )를 누릅니다.

통계 데이터를 입력한 경우 통계 레지스터의 내용을 확인할 수 있습니다.

**MEM** **1** (**1VAR**) **ENTER** 를 누른 다음 **^** 및 **v**를 사용하여 통계 레지스터를 봅니다.

### 예: 통계 레지스터 보기

**Σ+**를 사용하여 데이터 쌍 (1,2)와 (3,4)를 통계 레지스터에 저장합니다. 그런 다음 저장된 통계 값을 봅니다.

키:

디스플레이:

설명:

**CLEAR** **4** (**4Σ**)

**2** **ENTER** **1** **Σ+**

2.0000  
1.0000

통계 레지스터를 지웁니다.

첫 번째 데이터 쌍 (1,2)를 저장합니다.

**4** **ENTER** **3** **Σ+**

4.0000  
2.0000

두 번째 데이터 쌍 (3,4)를 저장합니다.

n=

↑ VAR 카탈로그를 표시하고 n 레지스터를 봅니다.

**MEM** **1** (**1VAR**)

2.0000

**^**

$\Sigma xy =$   
14.0000

↑  $\Sigma xy$  레지스터를 봅니다.

**^**

$\Sigma y^2 =$   
20.0000

↓

↑  $\Sigma y^2$  레지스터를 봅니다.

↓

## 12-12 통계 연산

⤴	$\Sigma x^2 =$ 10.0000	⬆ $\Sigma x^2$ 레지스터를 봅니다. ⬇
⤴	$\Sigma y =$ 6.0000	⬆ $\Sigma y$ 레지스터를 봅니다. ⬇
⤴	$\Sigma x =$ 4.0000	⬆ $\Sigma x$ 레지스터를 봅니다. ⬇
⤴	$n =$ 2.0000	⬆ $n$ 레지스터를 봅니다. ⬇
ⓐ	4.0000 2.0000	VAR 카탈로그를 나갑니다.

## 통계 레지스터 액세스

다음 표에는 HP 35s의 통계 레지스터 지정이 표시되어 있습니다. 합계 레지스터는 수식, 방정식 및 프로그램에서 번호가 아니라 이름으로 참조해야 합니다.

### 통계 레지스터

레지스터	번호	설명
$n$	-27	누적된 데이터 쌍의 개수
$\Sigma x$	-28	누적된 $x$ 값의 합계
$\Sigma y$	-29	누적된 $y$ 값의 합계
$\Sigma x^2$	-30	누적된 $x$ 값의 제곱 합계
$\Sigma y^2$	-31	누적된 $y$ 값의 제곱 합계
$\Sigma xy$	-32	누적된 $x$ 값과 $y$ 값의 곱 합계

원하는 레지스터의 번호(-27 - -32)를  $I$  또는  $J$ 에 저장한 다음 합계(값  $\text{STO} \text{ (I)}$  또는  $\text{(J)}$ )를 저장하여 합계가 포함된 통계 레지스터를 로드할 수 있습니다. 마찬가지로,  $\text{VIEW} \text{ (I)}$  또는  $\text{(J)}$  (또는  $\text{RCL} \text{ (I)}$  또는  $\text{(J)}$ )를 눌러 레지스터 값을 보거나 다시 호출할 수 있습니다. 디스플레이에 레지스터 이름이 포함된 레이블이 지정됩니다. SUMS 메뉴에는 레지스터 값을 다시 호출하는 기능이 있습니다. 자세한 내용은 14장의 “변수 및 레이블에 간접 주소 지정”을 참조하십시오.



# 2부

## 프로그래밍

---



## 단순 프로그래밍

이 설명서의 1부에서는 수동으로 각 개별 연산의 키를 눌러 사용할 수 있는 기능과 연산을 소개했습니다. 그리고 매번 모든 키 입력을 수행하지 않고 방정식을 사용하여 계산을 반복할 수 있는 방법을 알아보았습니다.

2부에서는 반복 계산, 즉 추가 입출력 제어나 보다 복잡한 논리가 관련될 수 있는 계산에 *프로그램*을 사용하는 방법을 알아보겠습니다. 프로그램을 통해 원하는 대로 정확하게 연산과 계산을 반복할 수 있습니다.

이 장에서는 일련의 연산을 프로그래밍하는 방법을 알아보겠습니다. 다음 장인 “프로그래밍 기술”에서는 서브루틴과 조건부 명령에 대해 알아보겠습니다.

### 예: 단순 프로그래밍

반지름이 5인 원의 넓이를 구하려면

$A = \pi r^2$  공식을 사용하고

**RPN 모드:** 5  $x^2$   $\leftarrow$   $\pi$   $\times$

**ALG 모드:** 5  $y^x$  2  $\times$   $\leftarrow$   $\pi$   $\text{ENTER}$

를 눌러 이 원의 결과인 78.5398을 구합니다.

그러나 다른 많은 원의 *넓이*를 구하려면 어떻게 하겠습니까?

다른 반지름을 나타내기 위해 “5”만 변경하여 매번 제공된 키 입력을 반복하는 대신 반복 가능한 키 입력을 프로그램에 넣을 수 있습니다.

**RPN 모드**  
 0001  $\times 2$   
 0002 P  
 0003  $\times$

**ALG 모드**  
 0001 SQ(x) $\times$ P

이 간단한 프로그램에서는 프로그램 실행이 시작될 때 반지름의 값이 X 레지스터 (디스플레이)에 있다고 가정합니다. 이 프로그램은 넓이를 계산하고 X 레지스터에 유지합니다.

RPN 모드에서 이 프로그램을 프로그램 메모리에 입력하려면 다음을 수행합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
[] [CLEAR] [3]		메모리를 지웁니다.
(3ALL) [←] (Y) [ENTER]		
[] [PRGM]		프로그램 입력 모드를 활성화합니다 (PRGM 표시기가 켜짐).
[GTO] [.] [.]	PRGM TOP	프로그램 포인터를 PRGM TOP으로 재설정합니다.
[] [x <sup>2</sup> ]	0001 $\times 2$	(반지름) <sup>2</sup>
[←] [π]	0002 P	
[×]	0003 $\times$	$\frac{1}{2} \pi r^2 = \pi r^2$
[] [PRGM]		프로그램 입력 모드를 종료합니다.

이 프로그램을 실행하여 반지름이 5인 원의 넓이를 구합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
[GTO] [.] [.]		프로그램이 시작 부분으로 설정됩니다.
5 [R/S]	78.5398	결과입니다.

ALG 모드에서 이 프로그램을 프로그램 메모리에 입력하려면 다음을 수행합니다.

키:	디스플레이:	설명:
<b>(ALG 모드)</b>		
[] [CLEAR] [3]		메모리를 지웁니다.
(3ALL) [←] (Y) [ENTER]		

## 13-2 단순 프로그래밍



 PRGM

프로그램 입력 모드를 활성화합니다 (PRGM 표시기가 켜짐).  
프로그램 포인터를 PRGM TOP으로 재설정합니다.

PRGM TOP

  RCL   

0001 SQ(X)×P









$넓이 = \pi x^2$



 PRGM

프로그램 입력 모드를 종료합니다.

이 프로그램을 실행하여 반지름이 5인 원의 넓이를 구합니다.

키:	디스플레이:	설명:
(ALG 모드)		
  		프로그램이 시작 부분으로 설정됩니다.
   	5→X	5를 X에 저장합니다.
	5.0000 78.9358	결과입니다.

원의 넓이에 대한 위의 프로그램을 계속 사용하여 프로그래밍 개념과 방법을 설명하겠습니다.

## 프로그램 디자인

다음 항목에서는 프로그램에 넣을 수 있는 명령을 보여 줍니다. 프로그램에 넣는 내용에 따라 프로그램을 볼 때 표시되는 모양과 실행 시의 작동 방식이 달라집니다.

### 모드 선택

RPN 모드에서 만들어 저장한 프로그램은 RPN 모드로 편집 및 실행해야 하고, ALG 모드에서 만들어 저장한 프로그램이나 단계는 ALG 모드로 편집 및 실행해야 합니다. 그렇지 않으면 결과가 잘못될 수 있습니다.

## 프로그램 경계(LBL 및 RTN)

둘 이상의 프로그램을 프로그램 메모리에 저장하려는 경우 시작 부분을 표시하는 *레이블*(예: R001 LBL R) 및 끝을 표시하는 *리턴*(예: R005 RTN)이 필요합니다. 줄 번호에는 레이블과 일치하는 R가 지정됩니다.

### 프로그램 레이블

프로그램 및 *루틴*이라고 하는 프로그램의 세그먼트는 레이블로 시작해야 합니다. 레이블을 기록하려면 다음을 누릅니다.

 **LBL** 문자 키

레이블은 A에서 Z 사이의 단일 문자입니다. 문자 키는 3장에서 설명된 대로 변수에 사용되는 것과 동일하게 사용됩니다. 동일한 레이블을 여러 번 지정할 수는 없지만(DUPLICAT·LBL이라는 메시지가 표시됨), 레이블과 변수가 동일한 문자를 사용할 수 있습니다.

하나의 프로그램(최상위 프로그램)은 레이블 없이 메모리에 배치될 수 있습니다. 그러나 인접한 프로그램은 서로 구분되도록 사이에 레이블이 있어야 합니다.

프로그램에 포함될 수 있는 최대 줄 수는 999줄입니다.

### 프로그램 리턴

프로그램과 서브루틴은 리턴 명령으로 끝나야 합니다. 키 입력은 다음과 같습니다.

 **RTN**

프로그램 실행이 완료되면 마지막 RTN 명령이 프로그램 포인터를 프로그램 메모리의 맨 위인 PRGM TOP으로 되돌립니다.

## 프로그램에 RPN, ALG 및 방정식 사용

키보드에서 계산하는 것과 동일한 방법으로 프로그램에서 계산할 수 있습니다.

- RPN 연산 사용(2장에서 설명한 것처럼 스택에서 작동)
- ALG 연산 사용(부록 C에서 설명)
- 방정식 사용(6장에서 설명)

앞의 예에서는 일련의 *RPN 연산*을 사용하여 원의 넓이를 계산했습니다. 대신 프로그램에 *방정식*을 사용할 수 있습니다. 이 장의 뒷부분에 예가 나와 있습니다. 대부분의 프로그램은 RPN 및 방정식의 조합으로, 둘의 장점을 사용합니다.

### RPN 연산의 장점

메모리를 덜 사용합니다.  
실행 속도가 빠릅니다.

### 방정식 및 ALG 연산의 장점

쓰고 읽기가 쉽습니다.  
자동으로 프롬프트를 표시할 수 있습니다.

프로그램에서 방정식이 포함된 줄을 실행하면 **[XEQ]**가 방정식 목록의 방정식을 계산하는 것과 동일한 방식으로 방정식이 계산됩니다. 프로그램 방정식에서 방정식의 “=”는 기본적으로 “~”로 처리됩니다. 방정식을 수식으로 작성한 다음 STO를 사용하여 값을 변수에 저장하는 것 이외의 지정 방정식에서 **[ENTER]** 대신 사용할 수 있는 프로그램 가능 기능은 없습니다.

두 가지 계산 모두, RPN 명령을 포함하여 입력, 출력 및 프로그램 흐름을 제어할 수 있습니다.

## 데이터 입력 및 출력

둘 이상의 입력이 필요하거나 둘 이상의 출력을 반환하는 프로그램의 경우 프로그램에서 정보를 입력하고 반환하는 방법을 결정할 수 있습니다.


입력의 경우 INPUT 명령으로 변수를 묻는 메시지를 표시하거나, 방정식에서 변수를 묻는 프롬프트를 표시하도록 하거나, 사전에 스택에 입력된 값을 사용할 수 있습니다.

출력의 경우 VIEW 명령으로 변수를 표시하거나, 방정식에서 파생된 메시지를 표시하거나, 줄 1에 프로세스를 표시하거나, 줄 2에 프로그램 결과를 표시하거나, 표시되지 않은 값을 스택에 유지할 수 있습니다.





이러한 내용에 대해서는 이 장의 뒷부분에 있는 “데이터 입력 및 표시”에서 설명합니다.





---


## 프로그램 입력







 **PRGM** 을 누르면 계산기가 프로그램 입력 모드로 전환되거나 다른 모드로 전환됩니다. 즉, **PRGM** 표시기가 켜지거나 꺼집니다. 프로그램 입력 모드의 키 입력은 메모리에 프로그램 줄로 저장됩니다. 각 명령이나 수식이 하나의 프로그램 줄을 사용합니다. ALG 모드에서는 프로그램에 직접 수식을 입력할 수 있습니다.

### 프로그램을 메모리에 입력하려면

1.  **PRGM** 을 눌러 프로그램 입력 모드를 활성화합니다.
2.    을 눌러 **PRGM TOP** 을 표시합니다. 이렇게 하면 *프로그램 포인터*가 다른 프로그램 앞의 알려진 위치로 설정됩니다. 프로그램 줄을 입력하면 다른 모든 프로그램 줄 *앞에* 삽입됩니다.

메모리에 있을 수 있는 다른 프로그램이 필요하지 않은 경우  **CLEAR**  (**3PGM**)을 눌러 프로그램 메모리를 지웁니다. *모든* 프로그램 삭제를 확인하려면 **CLR PGMS? Y\_N** 메시지가 표시된 후  (**Y**)  를 누릅니다.

3. 프로그램에 A에서 Z 사이의 단일 문자인 *레이블*을 지정합니다.  **LBL** 문자를 누릅니다. “*넓이*”를 나타내는 “A”와 같이 프로그램을 식별할 수 있는 문자를 선택합니다.

**DUPLICAT . LBL**이라는 메시지가 표시되면 다른 문자를 사용합니다. 대신 기존 프로그램을 지울 수 있습니다.  **MEM**  (**2PGM**)을 누르고  또는  를 눌러 레이블을 찾은 다음  **CLEAR** 및  를 누릅니다.

4. 계산기 연산을 프로그램 명령으로 기록하려면 수동으로 연산을 수행할 때와 동일한 키를 누릅니다. 많은 기능은 키보드에 표시되지 않으며 메뉴를 사용하여 액세스해야 합니다.

프로그램 줄에 방정식을 입력하려면 아래 명령을 참조하십시오.

## 13-6 단순 프로그래밍

5. 리턴 명령으로 프로그램을 끝냅니다. 이렇게 하면 프로그램이 실행된 후 프로그램 포인터가 다시 PRGM TOP으로 설정됩니다. **[RTN]**을 누릅니다.
6. **[C]** 또는 **[PRGM]**을 눌러 프로그램 입력을 취소합니다.

프로그램 줄의 숫자는 입력한 대로 정확하게 저장되며, ALL 또는 SCI 형식으로 표시됩니다. 긴 숫자가 디스플레이에서 축약되는 경우 **[SHOW]**를 눌러 모든 숫자를 표시합니다.

### 프로그램 줄에 방정식을 입력하려면

1. **[EQN]**을 눌러 방정식 입력 모드를 활성화합니다. EQN 표시기가 켜집니다.
2. 방정식 목록에서와 동일한 방식으로 방정식을 입력합니다. 자세한 내용은 6장을 참조하십시오. 입력 시 오류를 수정하려면 **[←]**를 사용합니다.
3. **[ENTER]**를 눌러 방정식을 종료하고 왼쪽 끝을 표시합니다. 방정식이 방정식 목록에 포함되지 *않습니다*.

방정식을 입력한 후 **[SHOW]**를 눌러 해당 체크섬과 길이를 확인할 수 있습니다. 값을 디스플레이에 유지하려면 **[SHOW]** 키를 계속 누르고 있으면 됩니다.

긴 방정식의 경우 **[→]** 및 **[←]** 표시기가 나타나 이 프로그램 줄에 대해 스크롤이 활성화되었음을 표시합니다. **[←]** 및 **[→]**를 사용하여 디스플레이를 스크롤할 수 있습니다.

### 지우기 기능 및 백스페이스 키

프로그램 입력 중 다음과 같은 특수 상태를 확인합니다.

- **[C]**는 항상 프로그램 입력을 취소하며 숫자를 0으로 지우지 않습니다.
- 프로그램 줄 보기 상태에서 **[←]**는 현재 프로그램 줄을 삭제하고 **[←]/[→]**는 편집 상태를 시작합니다. 프로그램 줄 편집 상태에서 **[←]**는 커서 앞의 한 문자를 삭제합니다.
- X 레지스터를 지우는 기능을 *프로그램*하려면 **[CLEAR]** **[1]** (1×)를 사용합니다.

프로그램에 줄을 삽입하거나 지우면 필요한 경우 자동으로 GTO 및 XEQ 문이 업데이트됩니다.

예:


R001 LBL A  
R002 2+3  
R003 1+2  
R004  
GTO R003

이제 줄 A002를 지우면 줄 A004가 "A003 GTO A002"로 변경됩니다.



## 프로그램의 기능 이름

프로그램 줄에 사용되는 기능 이름은 키, 메뉴 또는 방정식의 함수 이름과 같지 않<sup>아</sup>도 됩니다. 일반적으로 프로그램에 사용되는 이름은 키나 메뉴에 사용할 수 있는 이름보다 더 축약된 이름입니다.

### 예: 레이블이 지정된 프로그램 입력

다음 키 입력은 원의 넓이에 대한 이전 프로그램을 삭제하고 레이블 및 리턴 명령을 포함하는 새 프로그램을 입력합니다. 입력 중에 실수를 한 경우 를 눌러 현재 프로그램 줄을 삭제한 다음 줄을 올바르게 다시 입력합니다.

키:	디스플레이:	설명:
(RPN 모드)		
 PRGM		프로그램 입력 모드를 활성화 화합니다(PRGM 이 켜짐). 프로그램 메모리를 모두 지웁니다.
 CLEAR <b>3</b> (3PGM)  (Y)	PRGM TOP	
ENTER		
 LBL <b>A</b>	R001 LBL A	이 프로그램에 "넓이"를 나타내는 루틴 A라는 레이 블을 지정합니다. 3개의 프로그램 줄을 입력 합니다.
 $x^2$	R002 $\times 2$	
 $\pi$	R003 P	
 $\times$	R004 $\times$	
 RTN	R005 RTN	프로그램을 끝냅니다.

 **MEM** **2** (2PGM) LBL A  
 LN=15  
 **SHOW** CK=DAF1  
 LN=15











**C** **C**

레이블 A와 프로그램 길이 (바이트 단위)를 표시합니다. 체크섬 및 프로그램 길이입니다. 프로그램 입력을 취소합니다 (PRGM 표시기가 꺼짐).

체크섬이 다른 경우 여기에 제공된 대로 정확하게 프로그램을 입력하지 않은 것입니다.

### 예: 방정식이 있는 프로그램 입력

다음 프로그램에서는 이전 프로그램과 같이 RPN 연산을 사용하는 대신 방정식을 사용하여 원의 넓이를 계산합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
 <b>PRGM</b>  <b>GTO</b> <b>.</b>	PRGM TOP	프로그램 입력 모드를 활성화하고 포인터를 메모리의 맨 위로 설정합니다. 이 프로그램에 “방정식”을 나타내는 루틴 E라는 레이블을 지정합니다.
		
 <b>LBL</b> <b>E</b>	E001 LBL E	반지름을 변수 R에 저장합니다.
 <b>STO</b> <b>R</b>	E002 STO R	
<b>EQN</b>  <b><math>\pi</math></b>		방정식 입력 모드를 선택하고, 방정식을 입력하고, 프로그램 입력 모드로 돌아갑니다.
<b>x</b> <b>RCL</b> <b>R</b>		
<b>y<sup>x</sup></b> <b>2</b> <b>ENTER</b>	E003 P×R <sup>2</sup>	
 <b>SHOW</b>	CK=7E5B LN=5	프로그램을 끝냅니다. 레이블 E와 프로그램 길이 (바이트 단위)를 표시합니다. 체크섬 및 프로그램 길이입니다. 프로그램 입력을 취소합니다.
 <b>RTN</b>	E004 RTN	
 <b>MEM</b> <b>2</b> (2PGM)	LBL E LN=17	
 <b>SHOW</b>	CK=2073 LN=17	
<b>C</b> <b>C</b>		

# 프로그램 실행

프로그램을 실행하려면 프로그램 입력이 활성 상태가 아니어야 합니다(프로그램 줄 번호가 표시되지 않고 PRGM이 꺼짐). [C]를 누르면 프로그램 입력 모드가 취소됩니다.

## 프로그램 실행(XEQ)

[XEQ] 레이블을 눌러 해당 문자가 레이블로 지정된 프로그램을 실행합니다.

프로그램을 처음부터 실행하려면 [XEQ] 레이블 [ENTER]를 누릅니다. 예를 들어, [XEQ] [A] [ENTER]를 누릅니다. 디스플레이에 "XEQ R001"이 표시되고 레이블 A의 맨 위에서 실행이 시작됩니다.

[XEQ] 레이블 줄 번호(예: [XEQ] [A] [0] [0] [5])를 눌러 다른 위치에서 시작되는 프로그램을 실행할 수도 있습니다.

메모리에 프로그램이 하나뿐이면 포인터를 프로그램 줄의 맨 위로 이동한 후 [R/S](실행/중지) 키를 눌러 프로그램을 실행할 수도 있습니다. PRGM 표시기가 나타나고 프로그램이 실행되는 동안 B 표시기가 켜집니다.

필요한 경우 프로그램을 실행하기 전에 데이터를 입력합니다.

예:

A 및 E 레이블이 지정된 프로그램을 실행하여 반지름이 각각 5, 2.5 및  $2\pi$ 인 세 원의 넓이를 구합니다. A 또는 E를 실행하기 전에 반지름을 입력해야 합니다.

키:	디스플레이:	설명:
(RPN 모드) [5] [XEQ] [A] [ENTER]	RUNNING 78.5398	반지름을 입력한 다음 프로그램 A를 시작합니다. 결과 넓이가 표시됩니다.
[2] [.] [5] [XEQ] [E] [ENTER]	19.6350	프로그램 E를 사용하여 두 번째 원의 넓이를 계산합니다.
[2] [←] [π] [×] [XEQ] [A] [ENTER]	124.0251	세 번째 원의 넓이를 계산합니다.

## 13-10 단순 프로그래밍



## 프로그램 테스트

프로그램에 오류가 있는 것을 알고 있지만 오류 위치를 모르는 경우 단계별로 실행하여 프로그램을 테스트하는 것이 좋습니다. 사용하기 전에 긴 프로그램이나 복잡한 프로그램을 테스트하는 것도 바람직합니다. 한 번에 한 줄씩, 단계별로 실행하면 각 프로그램 줄이 실행될 때마다 결과를 볼 수 있으므로 올바른 결과를 알고 있는 알려진 데이터의 진행 상황을 확인할 수 있습니다.

1. 일반적인 실행의 경우 프로그램 입력이 활성화 상태가 아니어야 합니다(PRGM 표시기가 꺼짐).
2. 프로그램 포인터를 프로그램의 시작 부분(LBL 명령)으로 설정합니다. 이 명령은 실행을 시작하지 않고 프로그램 포인터를 이동합니다.
3. 를 누르고 있습니다. 이렇게 하면 현재 프로그램 줄이 표시됩니다. 를 놓으면 줄이 실행된 후 실행 결과가 표시됩니다(X 레지스터에 있음). 이전 줄로 이동하려면 를 누릅니다. 프로그램이 실행되지 않습니다.
4. 프로그램 포인터가 다음 줄로 이동합니다. 오류(잘못된 결과 발생)를 찾거나 프로그램 끝에 도달할 때까지 3단계를 반복합니다.

프로그램 입력 모드가 활성화 상태이면  또는 는 줄을 실행하지 않고 프로그램 포인터만 변경합니다. 프로그램 입력 중에 커서 키를 누르고 있으면 줄이 자동으로 롤링됩니다.

### 예: 프로그램 테스트

A 레이블이 지정된 프로그램을 단계별로 실행합니다. 테스트 데이터로 반지름 5를 사용합니다. 시작하기 전에 프로그램 입력 모드가 활성화 상태가 *아님*을 확인합니다.

키:	디스플레이:	설명:
(RPN 모드)		
<input type="checkbox"/> <b>GTO</b> <input type="checkbox"/> <b>A</b> <input type="checkbox"/> <b>ENTER</b>	5.0000	프로그램 카운터를 레이블 A로 이동합니다.
<input checked="" type="checkbox"/> (누르고 있음) (놓음)	A001 LBL A 5.0000	
<input checked="" type="checkbox"/> (누르고 있음) (놓음)	A002 ×2 25.0000	입력을 제공합니다.
<input checked="" type="checkbox"/> (누르고 있음) (놓음)	A003 P 3.1416	$\pi$ 의 값입니다.

☑ (누르고 있음) (놓음)	A004 × 78.5398	25π입니다.
☑ (누르고 있음) (놓음)	A005 RTN 78.5398	프로그램의 끝입니다. 결과가 맞습니다.

## 데이터 입력 및 표시

계산기의 변수는 데이터 입력, 중간 결과 및 최종 결과를 저장하는 데 사용됩니다. 3장에서 설명한 대로 변수는 A에서 Z 사이의 문자로 식별되지만 변수 이름과 프로그램 레이블은 아무 관계도 없습니다.

프로그램에서 다음과 같은 방법으로 데이터를 얻을 수 있습니다.

- 변수 값을 묻는 프롬프트가 표시되는 INPUT 명령에서(가장 편리한 방법입니다.)
- 스택에서(나중에 사용하기 위해 STO를 사용하여 값을 변수에 저장할 수 있습니다.)
- 이미 값이 저장되어 있는 변수에서
- 자동 방정식 프롬프트에서(플래그 11을 설정하여 활성화된 경우)(방정식을 사용하는 경우 이 방법도 편리합니다.)

프로그램에서 다음과 같은 방법으로 정보를 표시할 수 있습니다.

- 변수 이름과 값을 표시하는 VIEW 명령으로(가장 편리한 방법입니다.)
- 스택에서(X 레지스터와 Y 레지스터의 값만 볼 수 있음). (PSE를 사용하면 X 레지스터와 Y 레지스터를 1초 동안 볼 수 있습니다.)
- 표시된 방정식에서(플래그 10을 설정하여 활성화된 경우). (“방정식”은 일반적으로 메시지이며 실제 방정식이 아닙니다.)

다음 항목에서는 이러한 입력 및 출력 기술 중 일부에 대해 설명합니다.

## 데이터 입력에 INPUT 사용

INPUT 명령(**↵** **INPUT** 변수)은 실행 중인 프로그램을 중지하고 제공된 변수를 묻는 프롬프트를 표시합니다. 이 디스플레이에는 다음과 같은 변수의 기존 값이 포함됩니다.

```
R?  
0.0000
```

여기서

“R”은 변수 이름이고,  
“?”는 정보를 묻는 프롬프트이고,  
0.0000은 변수에 저장된 현재 값입니다.

**R/S**(**실행/중지**)를 눌러 프로그램을 다시 시작합니다. 그러면 입력한 값이 X 레지스터의 내용을 덮어쓰고 제공된 변수에 저장됩니다. 표시된 값을 변경하지 않은 경우 해당 값이 X 레지스터에 유지됩니다.

INPUT 명령을 사용한 원의 넓이 프로그램은 다음과 같이 나타납니다.

RPN 모드	ALG 모드
A001 LBL A	A001 LBL A
A002 INPUT R	A002 INPUT R
A003 $\times 2$	A003 $SQ(R) \times P$
A004 P	A004 RTN
A005 $\times$	
A006 RTN	

### 프로그램에 INPUT 함수를 사용하려면

1. 필요한 데이터 값을 결정하고 이름을 지정합니다.  
원의 넓이 예에서 필요한 입력은 R에 지정할 수 있는 반지름뿐입니다.

2. 프로그램의 시작 부분에서 값이 필요한 각 변수에 대해 INPUT 명령을 삽입합니다. 프로그램의 뒷부분에서 제공된 값이 필요한 계산 부분을 작성할 때 **[RCL]** 변수 명령을 삽입하여 해당 값을 스택으로 다시 호출합니다.

또한 INPUT 명령은 방금 입력된 값을 X 레지스터에 유지하므로 나중에 변수를 다시 호출할 필요 없이 입력 후 필요할 때 사용할 수 있습니다. 이 방법을 사용할 경우 일부 메모리 공간을 절약할 수도 있습니다. 그러나 긴 프로그램에서는 모든 데이터를 먼저 입력한 다음 필요에 따라 개별 변수를 다시 호출하는 것이 더 간단합니다.

또한 프로그램이 중지되고 입력을 기다리는 동안 프로그램 사용자가 계산을 수행할 수 있습니다. 이 경우 스택의 내용이 변경되어 프로그램에서 수행할 다음 계산에 영향을 줄 수도 있습니다. 따라서 프로그램에서 INPUT 명령 전과 후의 X, Y 및 Z 레지스터 내용이 동일하다고 가정해서는 안 됩니다. 시작 시 모든 데이터를 수집한 후 계산에 필요할 때 다시 호출하는 경우 계산 직전에 스택의 내용이 변경되는 것을 방지할 수 있습니다.

### 프롬프트에 응답하려면

프로그램을 실행하면 각 INPUT에서 중지되고 해당 변수를 묻는 프롬프트가 표시됩니다(예: R?0.0000). 표시된 값과 X 레지스터의 내용이 R의 현재 내용이 됩니다.

- **숫자를 변경하지 않고 그대로 두려면 [R/S]**를 누릅니다.
- **숫자를 변경하려면** 새 숫자를 입력하고 **[R/S]**를 누릅니다. 새로운 숫자가 X 레지스터의 기존 값을 덮어씁니다. 필요한 경우 숫자를 분수로 입력할 수 있습니다. 숫자를 계산해야 하는 경우 일반적인 키보드 계산을 사용한 다음 **[R/S]**를 누릅니다. 예를 들어, RPN 모드에서 **[2] [ENTER] [5] [y<sup>x</sup>] [R/S]**를 누르거나 ALG 모드에서 **[2] [y<sup>x</sup>] [5] [ENTER] [R/S]**를 누를 수 있습니다. **[ENTER]**를 누르기 전에는 줄 2에 수식이 표시됩니다. **[ENTER]**를 누른 후에는 수식의 결과가 수식 대신 줄 2에 표시되고 X 레지스터에 저장됩니다.

- **INPUT 프롬프트를 취소하려면** **[C]**를 누릅니다. 변수의 현재 값이 X 레지스터에 유지됩니다. **[R/S]**를 눌러 프로그램을 다시 시작하면 취소된 INPUT 프롬프트가 반복됩니다. 숫자 입력 중에 **[C]**를 누르면 숫자가 0으로 지워집니다. INPUT 프롬프트를 취소하려면 **[C]**를 다시 누릅니다.

## 데이터 표시에 VIEW 사용

프로그래밍된 VIEW 명령(**[VIEW]** 변수)은 실행 중인 프로그램을 중지하며, 제공된 변수의 내용을 다음과 같이 표시하고 식별합니다.

R=  
78.5398

이 명령은 *표시 전용*이며, 숫자를 X 레지스터에 복사하지 않습니다. 분수 표시 모드가 활성 상태이면 값이 분수로 표시됩니다.

- **[ENTER]**를 누르면 이 숫자가 X 레지스터에 복사됩니다.
- 숫자가 14자보다 긴 경우(예: 2진수, 복소수, 벡터 숫자) **[▶]****[◀]** 및 **[▶]****[>]**를 누르면 나머지 숫자가 표시됩니다.
- **[C]**(또는 **[←]**)를 누르면 VIEW 디스플레이가 지워지고 X 레지스터가 표시됩니다.
- **[▶]** **[CLEAR]**를 누르면 표시된 변수의 내용이 지워집니다.

프로그램을 계속하려면 **[R/S]**를 누릅니다.

프로그램을 중지하지 않으려면 아래의 “중지하지 않고 정보 표시”를 참조하십시오.

예를 들어, 16장의 “정규 분포 및 역 정규 분포용” 프로그램을 참조하십시오. T 루틴의 끝에 있는 선 T015와 T016은 X의 결과를 표시합니다. 또한 이 프로그램의 VIEW 명령은 RCL 명령 뒤에 옵니다. RCL 명령이 필요하지는 않지만 VIEW 명령으로 확인된 변수를 X 레지스터로 가져와서 수동으로 계산이 가능하도록 하기 때문에 편리합니다. VIEW 디스플레이를 보는 동안 **[ENTER]**를 눌러도 동일한 효과가 있습니다. 16장과 17장의 다른 응용 프로그램에서도 VIEW 명령으로 확인된 변수를 X 레지스터에 유지합니다.

# 방정식을 사용하여 메시지 표시

방정식은 계산될 때까지 구문이 올바른지 확인되지 않습니다. 즉, 거의 모든 문자 시퀀스를 방정식과 같은 프로그램에 입력할 수 있으며, 임의의 방정식을 입력하듯이 입력하면 됩니다. 아무 프로그램 줄에서나 [EQN]을 눌러 방정식을 시작합니다. 숫자와 수학 키를 눌러 숫자와 기호를 입력합니다. 각 문자 앞에서 [RCL]을 누릅니다. [ENTER]를 눌러 방정식을 끝냅니다.

플래그 10을 설정하면 방정식이 계산되지 않고 표시됩니다. 즉, 방정식으로 입력한 모든 메시지를 표시할 수 있습니다. 플래그에 대해서는 14장에서 자세히 설명합니다.

메시지가 표시되면 프로그램이 중지됩니다. 실행을 계속하려면 [R/S]를 누릅니다. 표시된 메시지가 14자보다 길면 메시지를 표시할 때 ➡ 표시기가 켜집니다.

[F7] [←] 및 [F7] [→]를 사용하여 디스플레이를 스크롤할 수 있습니다.

프로그램을 중지하지 않으려면 아래의 “중지하지 않고 정보 표시”를 참조하십시오.

## 예: 프로그램의 INPUT, VIEW 및 메시지

반지름과 높이가 제공될 때 원통의 겉넓이와 부피를 구하는 방정식을 작성합니다. 프로그램에 (원통을 나타냄)라는 레이블을 지정하고, 변수 S(겉넓이), V(부피), R(반지름) 및 H(높이)를 사용합니다. 다음과 같은 공식을 사용합니다.

$$V = \pi R^2 H$$

$$S = 2\pi R^2 + 2\pi RH = 2\pi R(R + H)$$

키:	디스플레이:	설명:
(RPN 모드)		
[F7] [PRGM] [F7]		프로그램 입력으로, 프로그램 메모리를 지웁니다.
[CLEAR] [3] (3PGM)	PRGM TOP	
[←] [Y] [ENTER]		
[F7] [LBL] [C]	C001 LBL C	프로그램에 레이블을 지정합니다.
[←] [INPUT] [R]	C002 INPUT R	
[←] [INPUT] [H]	C003 INPUT H	반지름과 높이를 묻는 프롬프트를 표시하는 명령입니다.

## 13-16 단순 프로그래밍

**키:**  
**(RPN 모드)**  
 EQN  $\leftarrow$   $\pi$   $\times$   
 RCL R  $\times^x$  2  $\times$   
 RCL H ENTER

$\leftarrow$  SHOW

STO V  
 EQN 2  $\times$   $\leftarrow$   
 $\pi$   $\times$  RCL R  $\times$   
 ( ) RCL R +  
 RCL H ENTER

$\leftarrow$  SHOW

STO S  
 $\leftarrow$  FLAGS 1  
 (1SF)  $\cdot$  0  
 EQN RCL V  
 RCL O RCL L  
 $\rightarrow$  SPACE +  $\rightarrow$   
 SPACE RCL A  
 RCL R RCL E  
 RCL A ENTER

$\leftarrow$  FLAGS 1

(2CF)  $\cdot$  0  
 $\leftarrow$  VIEW V  
 $\leftarrow$  VIEW S  
 $\leftarrow$  RTN  
 $\leftarrow$  MEM 2

(2PGM)  
 $\leftarrow$  SHOW

C C

**디스플레이:**

C004 PXR^2xH

CK=74FE

LN=7

C005 STO V

C006 2xPxRx(R+  $\rightarrow$

CK=19B3

LN=11

C007 STO S

C008 SF 10

C009 VOL + ARE  $\rightarrow$

C010 CF 10

C011 VIEW V

C012 VIEW S

C013 RTN

LBL C

LN=67

CK=97C3

LN=67

**설명:**

부피를 계산합니다.

체크섬 및 방정식 길이입니다.

부피를 V에 저장합니다.

걸넓이를 계산합니다.

체크섬 및 방정식 길이입니다.

걸넓이를 S에 저장합니다.

플래그 10을 설정하여 방정

식을 표시합니다.

방정식에 메시지를 표시합

니다.

플래그 10을 지웁니다.

부피를 표시합니다.

걸넓이를 표시합니다.

프로그램을 끝냅니다.

레이블 C와 프로그램 길이

(바이트 단위)를 표시합니다.

체크섬 및 프로그램 길이입

니다.

프로그램 입력을 취소합니다.

이제 반지름이  $2\frac{1}{2}$ cm이고 높이가 8cm인 원통의 부피와 겉넓이를 구합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
<b>XEQ</b> <b>C</b> <b>ENTER</b>	R? 값	C실행을 시작하고 R을 묻는 프롬프트를 표시합니다. R에 있는 값을 표시합니다.
<b>2</b> <b>.</b> <b>1</b> <b>.</b> <b>2</b>	H?	$2\frac{1}{2}$ 를 분수로 입력합니다.
<b>R/S</b>	값	H를 묻는 프롬프트를 표시합니다.
<b>8</b> <b>R/S</b>	VOL + AREA	표시되는 메시지입니다.
<b>R/S</b>	V= 157.0796	$\text{cm}^3$ 단위의 부피입니다.
<b>R/S</b>	S= 164.9336	$\text{cm}^2$ 단위의 겉넓이입니다.

### 중지하지 않고 정보 표시

일반적으로 프로그램은 VIEW를 사용하여 변수를 표시하거나 방정식 메시지를 표시할 때 중지됩니다. 대체로 실행을 다시 시작하려면 **R/S**를 눌러야 합니다.

필요한 경우 정보가 표시되는 동안 프로그램이 계속되도록 할 수 있습니다. VIEW 명령이나 표시된 방정식 뒤의 다음 프로그램 줄에 PSE (일시 중지) 명령이 포함되어 있는 경우 정보가 표시되고 1초 중지된 후 실행이 계속됩니다. 이 경우 스크롤 또는 키보드 입력을 사용할 수 없습니다.

디스플레이는 플래그 7이 설정된 경우 RND 연산(분수로 반올림됨) 및 다른 디스플레이 작업에 의해 지워집니다.

**↔** **PSE**를 눌러 프로그램에 PSE를 입력합니다.

VIEW 및 PSE 줄이나 방정식 및 PSE 줄은 한 번에 한 줄씩 프로그램을 실행할 때 하나의 작업으로 처리됩니다.

### 13-18 단순 프로그래밍



---

## 프로그램 중지 또는 일시 중단

### 중지 또는 일시 중지(STOP, PSE) 프로그래밍

- 프로그램 입력 중에 **[R/S]**(*실행/중지*)를 누르면 STOP 명령이 삽입됩니다. 이 경우 X 레지스터의 내용이 표시되고, 키보드에서 **[R/S]**를 눌러 다시 시작할 때까지 실행 중인 프로그램이 중지됩니다. 프로그램 포인터를 메모리 맨 위로 되돌리지 않고 프로그램을 끝내려면 RTN 대신 STOP을 사용할 수 있습니다.
- 프로그램 입력 중에 **[PSE]**를 누르면 PSE(*일시 중지*) 명령이 삽입됩니다. 이 경우 실행 중인 프로그램이 일시 중단되고 X 레지스터의 내용이 1초 동안 표시됩니다. 단, 다음과 같은 경우는 예외입니다. PSE가 VIEW 명령이나 표시되는 방정식(플래그 10 설정) 바로 뒤에 오는 경우 변수나 방정식이 대신 표시되고 1초 중지된 후에도 디스플레이가 유지됩니다.

### 실행 중인 프로그램 일시 중단

언제든지 **[C]** 또는 **[R/S]**를 눌러 실행 중인 프로그램을 일시 중단할 수 있습니다. 프로그램은 현재 명령을 완료한 후 중지됩니다. 프로그램을 다시 시작하려면 **[R/S]**(*실행/중지*)를 누릅니다.

프로그램을 일시 중단한 후 **[XEQ]**, **[GTO]** 또는 **[RTN]**을 누르면 **[R/S]**를 사용하여 프로그램을 다시 시작할 수 *없습니다*. 대신 프로그램을 다시 실행합니다 (**[XEQ]** 레이블 줄 번호).

### 오류 중지

실행 중인 프로그램에서 오류가 발생하는 경우 프로그램 실행이 중지되고 오류 메시지가 디스플레이에 나타납니다. 부록 F에 메시지 및 상태 목록이 있습니다.






오류 발생 명령이 포함된 프로그램 줄을 확인하려면 **[PRGM]**을 누릅니다. 해당 위치에서 프로그램이 중지됩니다. 예를 들면, 잘못된 불능(0으로 나누기) 오류를 발생시키는 ÷ 명령 같은 것입니다.

---

## 프로그램 편집

프로그램 줄을 삽입, 삭제 및 편집하여 프로그램 메모리의 프로그램을 수정할 수 있습니다. 프로그램 줄에 방정식이 포함되어 있는 경우 방정식을 편집할 수 있습니다.

### 프로그램 줄을 삭제하려면







1. 관련된 프로그램이나 루틴을 선택하고  또는  를 눌러 변경해야 하는 프로그램 줄을 찾습니다. 스크롤을 계속하려면 커서 키를 계속 누르고 있으면 됩니다.
2. 변경하려는 줄을 삭제합니다. 직접  를 누릅니다(실행 취소 기능이 활성화 됨). 그러면 포인터가 *이전* 줄로 이동합니다. 두 줄 이상 연속되는 프로그램 줄을 삭제하는 경우 그룹의 *마지막* 줄에서 시작합니다.
3. 새 명령을 입력합니다(있는 경우). 이 명령이 삭제된 명령을 대체합니다.
4. 프로그램 입력을 종료합니다(  또는  ).

### 프로그램 줄을 삽입하려면










1. 줄을 삽입하려는 위치 *앞에* 있는 프로그램 줄을 찾아서 표시합니다.
2. 새 명령을 입력합니다. 현재 표시된 줄 *뒤에* 삽입됩니다.

예를 들어, 프로그램의 줄 A004와 줄 A005 사이에 새 줄을 삽입하려는 경우 먼저 줄 A004를 표시한 다음 명령을 입력합니다. 원래 줄인 A005부터 시작하여 이후의 프로그램 줄이 아래로 이동되고 적절하게 번호가 다시 매겨집니다.

### 프로그램 줄의 피연산자, 수식 또는 방정식을 편집하려면

1. 편집하려는 프로그램 줄을 찾거나 표시합니다.
2.  또는  를 눌러 프로그램 줄 편집을 시작합니다. 두 기능은 “\_” 편집 커서를 커지만 프로그램 줄의 내용은 삭제하지 않습니다.  
 키는 프로그램 줄의 왼쪽으로 커서를 활성화합니다.  
 키는 프로그램 줄의 끝으로 커서를 활성화합니다.
3. “\_” 커서를 이동하고 반복해서  를 눌러 불필요한 숫자나 기능을 삭제한 다음 프로그램 줄의 나머지 부분을 다시 입력합니다.  를 누르면 실행 취소 기능이 활성화됩니다.



## 참고:

1. 커서가 프로그램 줄에서 활성화 상태이면  또는  키를 사용할 수 없습니다.
2. 프로그램 줄을 편집 중이고(커서 활성화됨) 프로그램 줄이 비어 있으면  를 사용해도 아무 효과가 없습니다. 프로그램 줄을 지우려는 경우  를 누르면 프로그램 줄이 지워집니다.
3.   및   키를 사용하여 긴 프로그램 줄을 편집하지 않고 검토할 수 있습니다.
4. ALG 모드에서는  를 기능으로 사용할 수 없습니다. 프로그램 줄의 유효성을 검사하는 데 사용됩니다.
5. 입력 시 사용된 모드에 관계없이 아무 모드에서나 방정식을 편집할 수 있습니다.










---

## 프로그램 메모리

### 프로그램 메모리 보기

  을 누르면 계산기가 프로그램 입력 모드로 전환되거나(PRGM 표시기가 켜지고 프로그램 줄이 표시됨) 다른 모드로 전환됩니다. 프로그램 입력 모드가 활성화 상태이면 프로그램 메모리의 내용이 표시됩니다.

프로그램 메모리는 PRGM TOP에서 시작됩니다. 프로그램 줄 목록은 순환되므로 프로그램 포인터를 맨 아래에서 맨 위로, 또는 그 반대로 래핑할 수 있습니다. 프로그램 입력이 활성화 상태이면 다음 네 가지 방법으로 프로그램 포인터(표시된 줄)를 변경할 수 있습니다.

-   및   를 사용하면 한 레이블에서 다른 레이블로 이동할 수 있습니다. 레이블이 정의되지 않은 경우 프로그램의 맨 위나 맨 아래로 이동합니다.
- 한 번에 여러 줄을 이동하려면("스크롤")  또는  키를 누른 채로 있으면 됩니다.
-    을 눌러 프로그램 포인터를 PRGM TOP으로 이동합니다.

- **[GTO]** **[.]** 레이블 *nnn*을 눌러 특정 줄로 이동합니다.

프로그램 입력 모드가 활성 상태가 아닌 경우(프로그램 줄이 표시되지 않는 경우)

**[GTO]** 레이블 줄 번호를 눌러 프로그램 포인터를 이동할 수도 있습니다.

프로그램 입력 모드를 취소해도 프로그램 포인터의 위치는 변경되지 *않습니다*.

## 메모리 사용

프로그램 입력 중에 MEMORY FULL이라는 메시지가 표시되면 입력하려는 줄에 프로그램 메모리 공간이 부족한 것입니다. 프로그램이나 기타 데이터를 지워 추가 공간을 확보할 수 있습니다. 아래의 “한 개 또는 여러 개의 프로그램 지우기” 또는 부록 B의 “계산기 메모리 관리”를 참조하십시오.

## 프로그램 카탈로그(MEM)

프로그램 카탈로그는 각 레이블에 사용된 메모리의 바이트 수와 연관된 줄이 포함된 모든 프로그램 레이블 목록입니다. **[←]** **[MEM]** **[2]**(2PGM)을 눌러 카탈로그를 표시하고, **[↓]** 또는 **[↑]**를 눌러 목록 내에서 이동합니다. 이 카탈로그를 사용하여 다음을 수행할 수 있습니다.

- 프로그램 메모리에 있는 레이블과 레이블이 지정된 각 프로그램 또는 루틴의 메모리 사용을 검토합니다.
- 레이블이 지정된 프로그램을 실행합니다. 레이블이 표시되는 동안 **[XEQ]** 또는 **[R/S]**를 누릅니다.
- 레이블이 지정된 프로그램을 표시합니다. 레이블이 표시되는 동안 **[→]** **[PRGM]**을 누릅니다.
- 특정 프로그램을 삭제합니다. 레이블이 표시되는 동안 **[→]** **[CLEAR]**를 누릅니다.
- 제공된 프로그램 세그먼트와 연관된 체크섬을 확인합니다. **[←]** **[SHOW]**를 누릅니다.

레이블이 지정된 각 프로그램 세그먼트에서 사용하는 메모리 바이트 수가 카탈로그에 표시됩니다. 프로그램은 프로그램 레이블에 의해 식별됩니다.

## 13-22 단순 프로그래밍

LBL C  
LN=67

여기서 67은 프로그램에 사용되는 바이트 수입니다.

## 한 개 또는 여러 개의 프로그램 지우기

### 특정 프로그램을 메모리에서 지우려면

1. **[MEM]** **[2]** (2PGM) **[ENTER]** 를 누르고 프로그램 레이블을 표시합니다 (**[V]** 및 **[^]** 사용).
2. **[CLEAR]** 를 누릅니다.
3. **[C]** 를 눌러 카탈로그를 취소하거나 **[←]** 를 눌러 뒤로 이동합니다.

### 모든 프로그램을 메모리에서 지우려면

1. **[PRGM]** 을 눌러 프로그램 줄을 표시합니다(PRGM 표시기가 켜짐).
2. **[CLEAR]** **[3]** (3PGM) 을 눌러 프로그램 메모리를 지웁니다.
3. CLR PGMS? Y N이라는 메시지가 표시되어 확인을 요청합니다.  
**[<]** (Y) **[ENTER]** 를 누릅니다.
4. **[PRGM]** 을 눌러 프로그램 입력을 취소합니다.

모든 메모리를 지우면(**[CLEAR]** **[3]** (3ALL)) 모든 프로그램도 지워집니다.

## 체크섬

체크섬은 각 프로그램 레이블 및 연관된 줄(다음 레이블까지)에 제공되는 고유한 16진수 값입니다. 이 숫자는 프로그램 메모리에 입력한 기존 프로그램의 알려진 체크섬과 비교하는 데 유용합니다. 알려진 체크섬과 계산기에 표시된 체크섬이 같으면 모든 프로그램 줄이 올바르게 입력되었습니다. 체크섬을 확인하려면 다음을 수행합니다.

1. **[MEM]** **[2]** (2PGM) **[ENTER]** 를 눌러 프로그램 레이블 카탈로그를 표시합니다.
2. 필요한 경우 커서 키를 사용하여 해당 레이블을 표시합니다.
3. **[SHOW]** 를 눌러 유지한 채 **CK=체크섬** 및 **LN=길이**를 표시합니다.

예를 들어, 현재 프로그램("원통" 프로그램)의 체크섬을 확인하려면 다음을 수행합니다.

키:	디스플레이:	설명:
(RPN 모드)		
MEM  2	LBL C	67바이트를 사용하는 레이블
(2PGM)  ENTER	LN=67	C를 표시합니다.
SHOW (누르고 있음)	CK=97C3 LN=67	체크섬 및 길이

체크섬이 이 숫자와 일치하지 않는 경우 이 프로그램이 올바르게 입력되지 않았 습니다.

16장과 17장에 제공된 모든 응용 프로그램에는 프로그램 입력의 정확성을 확인 할 수 있도록 레이블이 지정된 각 루틴의 체크섬 값이 포함되어 있습니다.

또한 프로그램의 각 방정식에 체크섬이 있습니다. 이 항목의 앞부분에 나오는 "프로그램 줄에 방정식을 입력하려면"을 참조하십시오.


## 프로그래밍할 수 없는 기능

HP 35s의 다음 기능은 프로그래밍할 수 *없습니다*.

CLEAR  3 (3PGM)	GTO  .  .
CLEAR  3 (3ALL)	GTO  . 레이블 줄 번호
	MEM
, , ,	SHOW
PRGM	EQN
,	FDISP
UNDO	CLEAR  6 (6CLVAR×)

---

## BASE를 사용하여 프로그래밍

 **BASE**를 사용하여 진법 모드를 변경하는 명령을 프로그래밍할 수 있습니다. 이러한 설정은 키보드에서 실행된 기능과 동일한 방식으로 프로그램에서 작동합니다. 따라서 4가지 진법 중 하나로 숫자를 받아들여 임의의 진수로 산술을 수행하고 결과를 표시하는 프로그램을 작성할 수 있습니다.

10진수가 아닌 다른 진법을 사용하는 프로그램을 작성하는 경우 계산기의 현재 설정과 프로그램의 명령 둘 다에 진법 모드를 설정합니다.

### 프로그램에서 진법 모드 선택

프로그램의 시작 부분에 BIN, OCT 또는 HEX 명령을 삽입합니다. 프로그램이 완료될 때 계산기 설정이 10진수 모드로 돌아가도록 일반적으로 프로그램의 끝에 DEC 명령을 포함해야 합니다.

진법 모드를 변경하는 프로그램의 명령은 *프로그램 실행 중과 그 후에 입력이 해석되는 방법과 출력이 표시되는 모양을 결정하지만* 입력 시 프로그램 줄에는 영향을 주지 *않습니다*.

### 프로그램 줄에 입력된 숫자

프로그램 입력을 시작하기 전에 진법 모드를 설정합니다. 진법 모드의 현재 설정에 따라 프로그램 결과가 결정됩니다.

표시기를 통해 현재 설정된 진법을 확인할 수 있습니다. 10진수 모드와 10진수가 아닌 모드로 표시된 아래 프로그램 줄을 비교합니다. 10진수와 비 10진수는 계산기의 디스플레이에서 모두 왼쪽 맞춥니다.

#### 10진수 모드 설정:

```
      :  
      :  
      :  
PRGM  
R009 BIN
```

#### 2진수 모드 설정:

```
      :  
      :  
      :  
PRGM BIN  
R009 BIN
```

A010 10

10진수의 경우 기호 "d"를 생략할 수 있습니다.

A010 10b

2진수의 경우 진법 기호 "b"를 추가해야 합니다.

⋮

⋮

## 다항식 및 호너(Horner) 방법

다항식과 같은 일부 수식은 값을 구하기 위해 동일한 변수를 여러 번 사용합니다. 예를 들어,

$$Ax^4 + Bx^3 + Cx^2 + Dx + E$$

수식은 변수  $x$ 를 4번 사용합니다. RPN 연산을 사용하여 이러한 수식을 계산하는 프로그램은 변수에서 저장된  $x$  복사본을 반복하여 다시 호출할 수 있습니다.

### 예:

$5x^4 + 2x^3$ 에 대한 RPN 연산을 사용하여 프로그램을 작성한 다음  $x = 7$ 에 대해 계산합니다.

키:	디스플레이:	설명:
(RPN 모드)		
PRGM  GTO		
	PRGM TOP	
LBL  A	A001 LBL A	
INPUT  X	A002 INPUT X	
	A003 5	5
RCL  X	A004 RCL X	
	A005 4	
	A006 $y^x$	$x^4$
	A007 $x$	$5x^4$
RCL  X	A008 RCL X	
	A009 3	
	A010 $y^x$	$x^3$
	A011 2	



<b>X</b>	A012 ×	$2x^3$
<b>+</b>	A013 +	$5x^4 + 2x^3$
<b>RTN</b>	A014 RTN	
<b>MEM 2</b> (2PGM)	LBL A LN=46	46바이트를 사용하는 레이블 A를 표시합니다.
<b>SHOW</b>	CK=ER18 LN=46	체크섬 및 길이입니다.
<b>C C</b>		프로그램 입력을 취소합니다.

이제  $x = 7$ 에 대해 이 다항식을 계산합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
<b>XEQ A ENTER</b>	X?	x를 묻는 프롬프트를 표시합니다.
<b>7 R/S</b>	값 12,691.0000	결과입니다.

모든 방정식

$Ax^4 + Bx^3 + Cx^2 + Dx + E$ 에 대한 이 프로그램의 보다 일반적인 형태는 다음과 같습니다.

```

A001 LBL A
A002 INPUT A
A003 INPUT B
A004 INPUT C
A005 INPUT D
A006 INPUT E
A007 INPUT X
A008 RCL X
A009 RCL× A
A010 RCL+ B
A011 RCL× X
A012 RCL+ C
A013 RCL× X
A014 RCL+ D
A015 RCL× X
A016 RCL+ E
A017 RTN

```

체크섬 및 길이: 9E5E 51



## 프로그래밍 기술

---

13장에서는 프로그래밍의 기초에 대해 설명했습니다. 이 장에서는 보다 복잡하지만 유용한 기술에 대해 살펴보겠습니다.

- 서브루틴을 통해 특정 작업을 전담하는 프로그램 부분을 구분하고 레이블을 지정하여 프로그램을 단순화합니다. 또한 서브루틴을 사용하면 일련의 단계를 여러 번 수행해야 하는 프로그램이 단축됩니다.
- 조건부 명령(비교 및 플래그)을 사용하여 사용할 명령이나 서브루틴을 결정합니다.
- 카운터와 함께 루프를 사용하여 명령 세트를 특정 횟수만큼 실행합니다.
- 간접 주소 지정을 사용하여 동일한 프로그램 명령으로 다른 변수에 액세스합니다.

---

## 프로그램의 루틴

프로그램은 하나 이상의 *루틴으로 이루어져 있습니다*. 루틴은 특정 작업을 수행하는 기능 단위입니다. 복잡한 프로그램의 경우 작업을 그룹화하고 구분하는 루틴이 필요합니다. 이렇게 하면 프로그램 작성, 읽기, 이해 및 변경이 더 쉬워집니다.

일반적으로 루틴은 레이블에서 시작되고 RTN 또는 STOP과 같은 프로그램/루틴 실행을 중지하는 명령으로 끝납니다.

### 서브루틴 호출(XEQ, RTN)

*서브루틴*은 다른 루틴에서 호출(실행)되고 서브루틴이 완료될 때 동일한 루틴으로 돌아가는 루틴입니다.

- 계산기 메모리에 프로그램이 하나뿐인 경우 다양한 레이블에서 루틴을 구분할 수 있습니다. 계산기 메모리에 둘 이상의 프로그램이 있는 경우 주 프로그램 레이블의 루틴 부분이 특정 줄 번호에서 시작되도록 하는 것이 좋습니다.
- 서브루틴 자체에서 다른 서브루틴을 호출할 수 있습니다.

이 장의 순서도에서는 다음과 같은 표기법을 사용합니다.

**R005 GTO B001** → ①      프로그램 실행이 이 줄부터 ← ①("1부터")로 표시된 줄 번호까지 분기됩니다.

**B001 LBL B** ← ①      프로그램 실행이 → ①("1까지")로 표시된 줄 번호부터 이 줄까지 분기됩니다.

아래 예에서는 서브루틴을 호출하여 입력한 숫자의 부호를 변경하는 방법을 보여줍니다. 줄 **D003 XEQ E001**에 의해 루틴 D에서 호출된 서브루틴 E는 숫자의 부호를 변경합니다. 서브루틴 E는 줄 D004에서 결과를 저장하고 표시하기 위해 프로그램 실행을 다시 루틴 D로 보내는 RTN 명령으로 끝납니다. 아래 순서도를 참조하십시오.

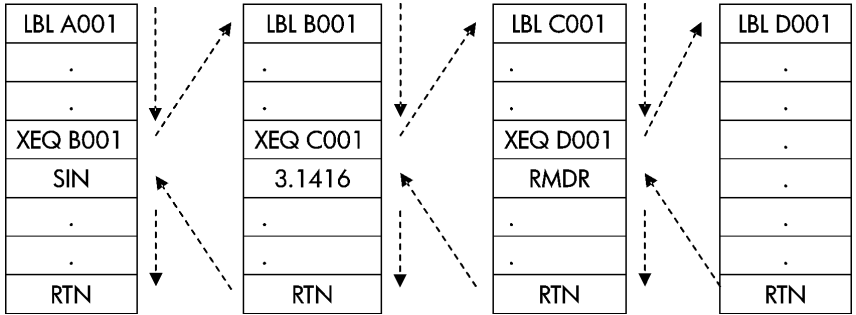
<b>D001 LBL D</b>		여기서부터 시작합니다.
<b>D002 INPUT X</b>		
<b>D003 XEQ E001</b>	→ ①	서브루틴 E를 호출합니다.
<b>D004 STO X</b>	← ②	여기서부터 돌아갑니다.
<b>D005 VIEW X</b>		
<b>D006 RTN</b>		
<b>E001 LBL E</b>	← ①	서브루틴을 시작합니다.
<b>E002 +/-</b>		숫자의 부호를 변경합니다.
<b>E003 RTN</b>	→ ②	루틴 D로 돌아갑니다.

## 중첩된 서브루틴

서브루틴은 다른 서브루틴을 호출할 수 있으며, 해당 서브루틴도 다른 서브루틴을 호출할 수 있습니다. 다른 서브루틴 내에서 서브루틴을 호출하는 이러한 서브루틴 "중첩"은 최상위 프로그램 수준을 제외하고 20개 서브루틴 수준의 스택으로 제한됩니다. 중첩된 서브루틴의 작업은 아래와 같이 수행됩니다.

MAIN 프로그램

(최상위)



프로그램의 끝

중첩 깊이가 20 수준을 초과하는 서브루틴을 실행하려고 하면 XEQ OVERFLOW 오류가 발생합니다.

**예: 중첩된 서브루틴**

S라는 레이블이 지정된 다음 서브루틴은

$$\sqrt{a^2 + b^2 + c^2 + d^2}$$

수식의 값을 큰 프로그램에 있는 큰 계산의 일부로 계산합니다. 이 서브루틴은 Q라는 레이블이 지정된 다른 서브루틴(중첩된 서브루틴)을 호출하여 반복되는 제곱 및 더하기 연산을 수행합니다. 이렇게 하면 서브루틴을 사용하지 않을 때보다 프로그램이 단축되어 메모리가 절약됩니다.

RPN 모드

S001 LBL S		여기서부터 서브루틴을 시작합니다.
S002 INPUT A		A를 입력합니다.
S003 INPUT B		B를 입력합니다.
S004 INPUT C		C를 입력합니다.
S005 INPUT D		D를 입력합니다.
S006 RCL D		데이터를 다시 호출합니다.
S007 RCL C		
S008 RCL B		
S009 RCL A		
S010 x2		$A^2$
S011 XEQ Q001 → ①		$A^2 + B^2$
② → S012 XEQ Q001 → ③		$A^2 + B^2 + C^2$
④ → S013 XEQ Q001 → ⑤		$A^2 + B^2 + C^2 + D^2$
⑥ → S014 $\sqrt{\quad} \times$		$\sqrt{A^2 + B^2 + C^2 + D^2}$
S015 RTN		주 루틴으로 돌아갑니다.
Q001 LBL Q	← ① ③ ⑤	중첩된 서브루틴입니다.
Q002 x<>y		
Q003 x2		
Q004 +		x2을 더합니다.
② ④ ⑥ ← Q005 RTN		서브루틴 S로 돌아갑니다.

## 분기(GTO)

서브루틴에서 확인했듯이 다음 줄이 아닌 프로그램의 특정 부분으로 실행을 전송해야 하는 경우가 있습니다. 이를 **분기**라고 합니다.

무조건 분기에서는 GTO(*이동*) 명령을 사용하여 특정 프로그램 줄(레이블 및 줄 번호)로 분기합니다.

### 프로그래밍된 GTO 명령

GTO 레이블 명령(**GTO** 레이블 줄 번호 *누름*)은 실행 중인 프로그램의 실행을 지정된 프로그램 줄로 전송합니다. 새 위치에서 프로그램 실행이 계속되고 자동으로 시작점으로 돌아가지 **않으므로** GTO는 서브루틴에는 사용되지 않습니다.

예를 들어, 16장의 “곡선 맞춤” 프로그램을 고려해 보십시오. **GTO Z 001** 명령은 세 독립 초기화 루틴 중 하나에서 프로그램의 핵심에 진입하는 공통 지점인 LBL Z 루틴으로 실행을 분기합니다.

S001 LBL S		여기서부터 시작할 수 있습니다.
.		
.		
S004 GTO Z001	→①	Z001로 분기합니다.
L001 LBL L		여기서부터 시작할 수 있습니다.
.		
.		
L004 GTO Z001	→①	Z001로 분기합니다.
E001 LBL E		여기서부터 시작할 수 있습니다.
.		
.		
E004 GTO Z001	→①	Z001로 분기합니다.
Z001 LBL Z	←①	여기까지 분기합니다.
.		
.		

## 키보드에서 GTO 사용

[GTO]를 사용하여 프로그램 실행을 시작하지 *않고* 프로그램 포인터를 지정된 레이블 줄 번호로 이동할 수 있습니다.

- PRGM TOP으로 이동: [GTO] [.] [.]
- 특정 줄 번호로 이동: [GTO] 레이블 줄 번호(줄 번호 < 1000). 예를 들면, [GTO] [.] [A] [0] [0] [5]와 같습니다. 예를 들어 [GTO] [A] [0] [0] [5]를 누르면 디스플레이에 "GTO R005"가 표시됩니다.
- 레이블의 첫 번째 줄로 이동(예: A001): [GTO] [A] [ENTER]를 누르고 있으면 디스플레이에 "GTO R001"이 표시됩니다.

---

## 조건부 명령

프로그램 실행 시퀀스를 변경하는 또 다른 방법은 두 숫자를 비교하고 명제가 거짓이면 다음 프로그램 명령을 건너뛰는 참/거짓 테스트인 *조건부 테스트*를 사용하는 것입니다.

예를 들어, 줄 A005의 조건부 명령이  $x=0?$  (즉, *x가 0과 같습니까?*)이면 프로그램에서 X 레지스터의 내용을 0과 비교합니다. X 레지스터에 0이 포함되어 *있으면* 프로그램이 다음 줄을 계속합니다. X 레지스터에 0이 포함되어 *있지 않으면* 프로그램이 다음 줄을 *건너뛰고* 줄 A007로 분기합니다. 일반적으로 이 규칙을 "참이면 수행"이라고 합니다.

	A001 LBL A		
	:		
	:		
	:		
참이면 다음 줄을 수행합니다.	A005 $x=0?$	→ ②	거짓이면 다음 줄을 건너뛵니다.
	① ← A006 GTO B001		
	A007 LN	← ②	
	A008 STO A		
	:		
	:		
	:		
	① → B001 LBL B		
	:		
	:		





위의 예에서는 조건부 테스트와 함께 사용되는 일반적인 기술을 보여 줍니다. “참”인 경우에만 실행되는 테스트 바로 뒤의 줄은 다른 레이블로의 분기입니다. 따라서 테스트의 순 효과는 특정 상황에서 다른 루틴으로 분기하는 것입니다.

다음 세 범주의 조건부 명령이 있습니다.

- 비교 테스트. X 레지스터와 Y 레지스터 또는 X 레지스터와 0을 비교합니다.
- 플래그 테스트. 설정 또는 해제될 수 있는 플래그 상태를 확인합니다.
- 루프 카운터. 일반적으로 지정된 횟수만큼 반복하는 데 사용됩니다.

## 비교 테스트(x?y, x?0)

프로그래밍에 사용할 수 있는 12개 비교가 있습니다.  **x?y** 또는  **x?0** 을 누르면 두 테스트 범주 중 하나의 메뉴가 표시됩니다.

- x와 y를 비교하는 테스트를 나타내는 x?y
- x와 0을 비교하는 테스트를 나타내는 x?0

x는 X 레지스터의 숫자를 나타내고, y는 Y 레지스터의 숫자를 나타냅니다. 테스트에서 변수 X와 Y를 비교하지는 않습니다. x?y 및 x?0을 사용하여 두 숫자를 비교할 수 있으며, 숫자 중 하나가 실수가 아닌 경우 **INVALID DATA**라는 오류 메시지가 반환됩니다.

비교 범주를 선택하고 원하는 조건부 명령의 메뉴 키를 누릅니다.

### 테스트 메뉴

x?y	x?0
$\neq(x \neq y?)$ 를 나타냄	$\neq(x \neq 0?)$ 를 나타냄
$\leq(\leq y?)$ 를 나타냄	$\leq(x \leq 0?)$ 를 나타냄
$<(x < y?)$ 를 나타냄	$<(x < 0?)$ 를 나타냄
$>(x > y?)$ 를 나타냄	$>(x > 0?)$ 를 나타냄
$\geq(x \geq y?)$ 를 나타냄	$\geq(x \geq 0?)$ 를 나타냄
$=(x = y?)$ 를 나타냄	$=(x = 0?)$ 를 나타냄

키보드에서 조건부 테스트를 실행하는 경우 계산기에 **YES** 또는 **NO**가 표시됩니다.

예를 들어,  $x=2$ 이고  $y=7$ 이면  $x < y$ 를 테스트합니다.

	<b>키:</b>	<b>디스플레이:</b>
<b>RPN 모드</b>	<b>7</b> <b>ENTER</b> <b>2</b> <b>←</b> <b>x?y</b> <b>&gt;</b> <b>&gt;</b> <b>(&lt;)</b> <b>ENTER</b>	YES
<b>ALG 모드</b>	<b>7</b> <b>x↔y</b> <b>2</b> <b>←</b> <b>x?y</b> <b>&gt;</b> <b>&gt;</b> <b>(&lt;)</b> <b>ENTER</b>	YES

**예:**

16장의 “정규 분포 및 역 정규 분포” 프로그램에서는 루틴 T에  $x < y$ ? 조건부를 사용합니다.

**프로그램 줄:  
(RPN 모드)**

**설명**




<pre> : : : T009 ÷ T010 STO+ X T011 ABS T012 0.0001 T013 x&lt;y? T014 GTO T001  T015 RCL X T016 VIEW X  : : : </pre>	<p><math>X_{guess}</math>의 수정을 계산합니다.</p> <p>수정을 더하여 새로운 <math>X_{guess}</math>를 구합니다.</p> <p>테스트를 통해 수정이 의미가 있는지 확인합니다.</p> <p>수정이 의미가 있는 경우 루프의 시작 부분으로 돌아갑니다.</p> <p>수정이 의미가 없는 경우 계속합니다.</p> <p>X의 계산된 값을 표시합니다.</p>
--	--

줄 T009에서는  $X_{guess}$ 의 수정을 계산합니다. 줄 T013에서는 계산된 수정의 절대 값을 0.0001과 비교합니다. 값이 0.0001보다 작으면(“참이면 수행”) 프로그램에서 줄 T014를 실행합니다. 값이 0.0001보다 크거나 같으면 프로그램이 줄 T015로 건너뜁니다.

## 플래그


플래그는 상태 표시기입니다. *설정(참)* 또는 *해제(거짓)*일 수 있습니다. *플래그 테스트*는 “참이면 수행” 규칙을 따르는 또 다른 조건부 테스트입니다. 테스트 플래그를 설정하면 프로그램 실행이 계속되고, 플래그를 지우면 한 줄을 건너뛸니다.

### 플래그 의미

HP 35s에는 0에서 11까지의 번호가 매겨진 12개의 플래그가 있습니다. 키보드나 프로그램 명령에서 모든 플래그를 설정하고 지우고 테스트할 수 있습니다. 12개 플래그의 기본 상태는 모두 *해제*입니다. 부록 B에 설명된 3 키 메모리 지우기 작업으로 모든 플래그가 지워집니다. 플래그는  **CLEAR** **3** (**3ALL**)  (Y) 의 영향을 받지 *않습니다*.

- **플래그 0, 1, 2, 3 및 4**는 미리 지정된 의미가 없습니다. 즉, 이러한 플래그의 상태는 제공된 프로그램에서 정의된 의미를 나타냅니다. 아래 예를 참조하십시오.
- **플래그 5**를 설정하면 프로그램 내에서 오버플로가 발생할 때 프로그램이 일시 중단되고 **OVERFLOW** 및 ▲가 표시됩니다. *오버플로*는 결과가 계산기에서 처리할 수 있는 가장 큰 숫자를 초과할 때 발생합니다. 오버플로 결과는 가능한 가장 큰 숫자로 대체됩니다. 플래그 5를 지우면 오버플로가 발생한 프로그램이 일시 중단되지 않습니다. 그러나 프로그램이 중지될 때 잠시 **OVERFLOW**가 표시됩니다.
- **플래그 6**은 오버플로 **TOO BIG**이 발생할 때마다 계산기에 의해 *자동으로* 설정됩니다(사용자가 직접 플래그 6을 설정할 수도 있음). 아무 효과도 없지만 이 플래그를 테스트할 수 있습니다. 뿐만 아니라 프로그램에 10진수가 아닌 진법을 사용하는 경우 프로그램의 **TOO BIG**에 대해서도 플래그 6이 설정됩니다.



플래그 5와 6을 사용하여 프로그램 중에 발생하는 오버플로 상태를 제어할 수 있습니다. 플래그 5를 설정하면 오버플로가 발생한 줄의 바로 뒷줄에서 프로그램이 중지됩니다. 프로그램에서 플래그 6을 테스트하면 오버플로가 발생할 때마다 프로그램의 흐름이나 결과를 변경할 수 있습니다.

- **플래그 7, 8 및 9**는 분수 표시를 제어합니다. 키보드에서 플래그 7을 제어할 수도 있습니다.  **FDISP**를 눌러 분수 표시 모드를 켜거나 끄면 플래그 7도 설정되거나 지워집니다.

플래그 상태	분수 제어 플래그		
	7	8	9
지우기 (기본값)	분수 표시를 끄고 실수를 현재 디스 플레이 형식으로 표시합니다.	분수 분모가 $\neq c$ 값 보다 크지 않습니 다.	분수를 가장 작은 형태로 줄입니다.
설정	분수 표시를 설정 하고 실수를 분수 로 표시합니다.	분수 분모가 $\neq c$ 값 의 인수입니다.	분수가 감소되지 않습니다. 플래그 8을 설정한 경우 에만 사용됩니다.

- **플래그 10**은 방정식의 프로그램 실행을 제어합니다.  
플래그 10을 지우면(기본 상태) 실행 중인 프로그램의 방정식이 계산되고  
결과가 스택에 유지됩니다.

플래그 10을 설정하면 실행 중인 프로그램의 방정식이 메시지로 표시되어  
VIEW 문처럼 동작합니다.

1. 프로그램 실행이 중지됩니다.
2. 프로그램 포인터가 다음 프로그램 줄로 이동합니다.
3. 스택에 영향을 주지 않고 방정식이 표시됩니다.  또는 를 눌러  
디스플레이를 지울 수 있습니다. 다른 키를 누르면 해당 키의 기능이  
실행됩니다.
4. 다음 프로그램 줄이 PSE 명령인 경우 1초 중지된 후 실행이 계속됩니다.  
플래그 10의 상태는 키보드에서의 SF 및 CF 작업 실행이나 프로그램의 S 및  
CF 문에 의해서만 제어됩니다.

- **플래그 11**은 프로그램의 방정식을 실행할 때 프롬프트를 제어하며, *키보드 실행 중의 자동 프롬프트에는 영향을 주지 않습니다.*

플래그 11을 지우면(기본 상태) 프로그램 방정식의 계산, SOLVE 및 ∫FN이 중단되지 않고 계속됩니다. 변수가 발견될 때마다 방정식에 있는 각 변수의 현재 값이 자동으로 다시 호출됩니다. INPUT 프롬프트는 영향을 받지 않습니다.

플래그 11을 설정하면 각 변수가 방정식에서 처음 발견될 때 변수를 묻는 프롬프트가 표시됩니다. 방정식에서 변수가 나타나는 횟수에 관계없이 변수를 묻는 프롬프트는 한 번만 발생합니다. 계산 시 미지수 변수에 대해 프롬프트가 발생하지 않고, 적분 시에는 적분 변수에 대해 프롬프트가 발생하지 않습니다. 실행 중지를 묻는 프롬프트를 표시합니다. **[R/S]**를 누르면 입력된 변수 값이나 **[R/S]**가 프롬프트에 대한 유일한 응답인 경우 표시된(현재) 변수 값을 사용하여 계산이 다시 시작됩니다.

플래그 11은 프로그램 방정식의 계산, SOLVE 또는 ∫FN 후에 자동으로 지워집니다. 플래그 11의 상태도 키보드에서의 SF 및 CF 작업 실행이나 프로그램의 SF 및 CF 문에 의해 제어됩니다.

### 설정된 플래그 표시기

플래그 0, 1, 2, 3 및 4는 해당 플래그가 설정되면 디스플레이에 켜지는 표시기가 있습니다. **0, 1, 2, 3** 또는 **4**가 있거나 없는 상태를 통해 언제든지 이러한 5개 플래그의 설정 여부를 확인할 수 있습니다. 그러나 플래그 5-11의 상태에 대해서는 이러한 표시가 없습니다. 이러한 플래그의 상태는 키보드에서 FS? 명령을 실행하여 확인할 수 있습니다. 아래의 “플래그 사용”을 참조하십시오.

### 플래그 사용

**[←] [FLAGS]**를 누르면 SF CF FS? 등의 FLAGS 메뉴가 표시됩니다.

원하는 기능을 선택하면 플래그 번호(0-11)를 묻는 프롬프트가 표시됩니다. 예를 들어, 플래그 0을 설정하려면 **[←] [FLAGS] [1](1SF) [0]**을 누르고, 플래그 10을 설정하려면 **[←] [FLAGS] [1](1SF) [•] [0]**을 누르고, 플래그 11을 설정하려면 **[←] [FLAGS] [1](1SF) [•] [1]**을 누릅니다.

## FLAGS 메뉴

메뉴 키	설명
SF <i>n</i>	<i>플래그를 설정합니다. 플래그 n을 설정합니다.</i>
CF <i>n</i>	<i>플래그를 지웁니다. 플래그 n을 지웁니다.</i>
FS? <i>n</i>	<i>플래그가 설정되었습니까? 플래그 n의 상태를 테스트합니다.</i>

플래그 테스트는 비교 테스트와 마찬가지로 프로그램 실행에 영향을 주는 조건부 테스트입니다. FS? *n* 명령은 제공된 플래그의 설정 여부를 테스트합니다. 설정된 경우 프로그램의 다음 줄이 실행됩니다. 설정되지 않은 경우 다음 줄을 건너뛰니다. 이것은 이 장의 앞부분에 있는 “조건부 명령”에서 설명한 “참이면 수행” 규칙입니다.

키보드에서 플래그를 테스트하는 경우 계산기에 “YES” 또는 “NO”가 표시됩니다. 테스트할 모든 조건이 알려진 상태로 시작되는지 프로그램에서 확인하는 것이 좋습니다. 현재 플래그 설정은 실행된 이전 프로그램에서 어떻게 설정되었는지에 따라 달라집니다. 예를 들어, 제공된 모든 플래그가 해제 상태이며, 프로그램에서 설정하는 경우에만 설정된다고 가정해서는 안 됩니다. 플래그를 설정할 수 있는 조건이 발생하기 전에 플래그를 지워 이러한 가정을 **확인**해야 합니다. 아래 예를 참조하십시오.

### 예: 플래그 사용

#### 프로그램 줄: (RPN 모드)

```
S001 LBL S
S002 CF 0
S003 CF 1
S004 INPUT X
S005 FS? 0
S006 LN
S007 STO X
S008 INPUT Y
S009 FS?1
S010 LN
S011 STO Y
```

#### 설명:

X의 표시기인 플래그 0을 지웁니다.  
Y의 표시기인 플래그 1을 지웁니다.  
X를 묻는 프롬프트를 표시하고 저장합니다.  
플래그 0이 설정된 경우...  
... X 입력의 자연 로그를 수행합니다.  
플래그 테스트 후 해당 값을 X에 저장합니다.  
Y를 묻는 프롬프트를 표시하고 저장합니다.  
플래그 1이 설정된 경우...  
... Y 입력의 자연 로그를 수행합니다.  
플래그 테스트 후 해당 값을 Y에 저장합니다.

S012 VIEW X                    값을 표시합니다.  
 S013 VIEW Y                    값을 표시합니다.  
 S014 RTN

체크섬 및 길이: 16B3 42

위에 표시된 대로 줄 S002 CF0과 S003 CF1을 작성하면 플래그 0과 1이 지워지므로 줄 S006과 S010에서 X 및 Y 입력의 자연 로그를 수행하지 않습니다.

줄 S002와 S003을 SF 0과 CF 1로 대체하면 플래그 0이 설정되므로 줄 S006에서 X 입력의 자연 로그를 수행합니다.

줄 S002와 S003을 CF 0과 SF 1로 대체하면 플래그 1이 설정되므로 줄 S010에서 Y 입력의 자연 로그를 수행합니다.

줄 S002와 S003을 SF 0과 SF 1로 대체하면 플래그 0과 1이 설정되므로 줄 S006과 S010에서 X 및 Y 입력의 자연 로그를 수행합니다.

위의 프로그램을 사용하여 플래그 사용 방법을 확인합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
[XEQ] [S] [ENTER]	X? <i>값</i>	레이블 S를 실행하고 X 값을 묻는 프롬프트를 표시합니다.
[1] [R/S]	Y? <i>값</i>	1을 X에 저장하고 Y 값을 묻는 프롬프트를 표시합니다.
[1] [R/S]	X=  1.0000	1을 X에 저장하고 플래그 테스트 후 X 값을 표시합니다.
[R/S]	Y=  1.0000	플래그 테스트 후 Y 값을 표시합니다.

다른 세 경우를 시도해 볼 수 있습니다. 시도 후에는 [↵] [FLAGS] [2] (2CF) [0] 및 [↵] [FLAGS] [2] (2CF) [1]을 눌러 플래그 1과 0을 지워야 합니다.

### 예: 분수 표시 제어

다음 프로그램을 통해 계산기의 분수 표시 기능을 사용할 수 있습니다. 프로그램에서 분수와 분모(/c 값)를 묻는 프롬프트를 표시하고 사용자 입력을 사용합니다.

또한 이 프로그램에는 세 개의 분수 표시 플래그(7, 8 및 9)와 “메시지 표시” 플래그(10)가 사용되는 방법의 예가 들어 있습니다.

이 프로그램의 메시지는 MESSAGE로 나열되며 방정식으로 입력됩니다.

1. **[EQN]**을 눌러 방정식 입력 모드를 설정합니다(EQN 표시기가 켜짐).
2. 메시지의 각 알파 문자에 대해 **[RCL]** 문자를 누르고, 각 공백 문자에 대해 **[SPACE]**를 누릅니다.
3. **[ENTER]**를 눌러 현재 프로그램 줄에 메시지를 삽입하고 방정식 입력 모드를 끝냅니다.

**프로그램 줄:  
(RPN 모드)**

**설명:**

F001	LBL F	분수 프로그램을 시작합니다.
F002	CF 7	세 개의 분수 플래그를 지웁니다.
F003	CF 8	
F004	CF 9	
F005	SF 10	메시지를 표시합니다.
F006	DEC	10진수를 선택합니다.
F007	INPUT V	숫자를 묻는 프롬프트를 표시합니다.
F008	INPUT D	분모(2 - 4095)를 묻는 프롬프트를 표시합니다.
F009	RCL V	메시지를 표시한 다음 10진수를 표시합니다.
F010	<b>DECIMAL</b>	
F011	PSE	
F012	STOP	
F013	RCL D	
F014	/C	/c 값을 설정하고 플래그 7을 설정합니다.
F015	RCL V	
F016	<b>MOST PRECISE</b>	메시지를 표시한 다음 분수를 표시합니다.
F017	PSE	
F018	STOP	
F019	SF 8	플래그 8을 설정합니다.
F020	<b>FACTOR DENOM</b>	메시지를 표시한 다음 분수를 표시합니다.
F021	PSE	
F022	STOP	
F023	SF 9	플래그 9를 설정합니다.
F024	<b>FIXED DENOM</b>	메시지를 표시한 다음 분수를 표시합니다.
F025	PSE	
F026	STOP	
F027	GTO F001	프로그램의 시작 부분으로 이동합니다.

체크섬 및 길이 : BE54 123



위의 프로그램을 사용하여 다른 형태의 분수 디스플레이를 확인합니다.

키:	디스플레이:	설명:
(RPN 모드)		
<b>XEQ</b> <b>F</b> <b>ENTER</b>	V?	레이블 F를 실행하고 분수(V)를
<b>2</b> <b>.</b> <b>5</b> <b>3</b> <b>R/S</b>	값	문는 프롬프트를 표시합니다.
<b>1</b> <b>6</b> <b>R/S</b>	D?	2.53을 V에 저장하고 분모(D)를
<b>R/S</b>	값	문는 프롬프트를 표시합니다.
<b>R/S</b>	DECIMAL	16을 /C 값으로 저장합니다. 메시
<b>R/S</b>	16.0000	지를 표시한 다음 10진수를 표시합
<b>R/S</b>	2.5300	니다.
<b>R/S</b>	MOST PRECISE	메시지에 분수 형식(분모가 16보다
<b>R/S</b>	2 8/15 ▼	큼)이 표시된 다음 분수가 표시됩니
<b>R/S</b>	2 8/15	다. ▼는 분자가 8보다 “조금 작음”
<b>R/S</b>	FACTOR DENOM	을 나타냅니다.
<b>R/S</b>	2 1/2 ▲	메시지에 분수 형식(분모가 16의 인
<b>R/S</b>	2 1/2	수임)이 표시된 다음 분수가 표시됩
<b>R/S</b>	FIXED DENOM	니다.
<b>R/S</b> <b>C</b> <b>←</b> <b>FLAGS</b>	2 8/16 ▲	메시지에 분수 형식(분모가 16임)이
<b>2</b> (2CF) <b>.</b> <b>0</b>	2.5300	표시된 다음 분수가 표시됩니다.
	2.5300	프로그램을 중지하고 플래그 10을
		지웁니다.

---

## 루프

뒤로, 즉 이전 줄의 레이블로 분기하면 프로그램의 일부를 여러 번 실행할 수 있습니다. 이를 *루프*라고 합니다.

```
D001 LBL D
D002 INPUT M
D003 INPUT N
D004 INPUT T
D005 GTO D001
```

이 루틴은 *무한 루프*의 예입니다. 이 루틴을 사용하여 초기 데이터를 수집할 수 있습니다. 세 값을 입력한 후 다른 루틴을 실행하기 위해 **[XEQ]** 레이블 줄 번호를 눌러 수동으로 이 루프를 일시 중단할 수 있습니다.

## 조건부 루프(GTO)

특정 조건이 충족될 때까지 작업을 수행하려고 하지만 루프를 반복해야 하는 횟수를 모르는 경우 조건부 테스트와 GTO 명령을 사용하여 루프를 만들 수 있습니다.

예를 들어, 다음 루틴에서는 루프를 사용하여 결과 *A*가 *B*보다 작거나 같을 때까지 값 *A*를 상수 값 *B*만큼 줄입니다.



### 프로그램 줄: (RPN 모드)

### 설명:

S001 LBL S	
S002 INPUT A	
S003 INPUT B	
S004 RCL A	스택의 A 위치를 기억하는 것보다 A를 다시 호출하는 것이 더 쉽습니다.
S005 RCL- B	$A - B$ 를 계산합니다.
S006 STO A	기존 A를 새 결과로 대체합니다.
S007 RCL B	비교를 위해 상수를 다시 호출합니다.
S008 <<v?	$B < \text{새로운 A}$ 입니까?
S009 GTO S004	예: 루프를 통해 빠기를 반복합니다.
S010 VIEW A	아니오: 새로운 A를 표시합니다.
S011 RTN	

체크섬 및 길이 : 2737 33

## 카운터를 사용하여 루프(DSE, ISG)

특정 횟수만큼 루프를 실행하려는 경우  **ISG** (증가, 크면 건너뛰) 또는  **DSE** (감소, 작거나 같으면 건너뛰) 조건부 기능 키를 사용합니다. 프로그램에서 루프 함수가 실행될 때마다 변수에 저장된 카운터 값이 자동으로 증가하거나 감소합니다. 루프 함수는 현재 카운터 값을 최종 카운터 값과 비교한 다음 결과에 따라 루프를 계속하거나 종료합니다.

카운트 다운 루프의 경우  **DSE** 변수를 사용합니다.

카운트 업 루프의 경우  **ISG** 변수를 사용합니다.



이러한 함수는 BASIC의 FOR-NEXT 루프와 동일한 기능을 수행합니다.

FOR 변수 = 초기 값 TO 최종 값 STEP 증가

.  
. .  
.

NEXT 변수

DSE 명령은 음수 증가의 FOR-NEXT 루프와 같습니다.

ISG 또는 DSE에 해당하는 shift 키를 누르면( **ISG** 또는  **DSE**), 아래에서 설명하는 루프 제어 번호가 포함될 변수를 묻는 프롬프트가 표시됩니다.

### 루프 제어 번호

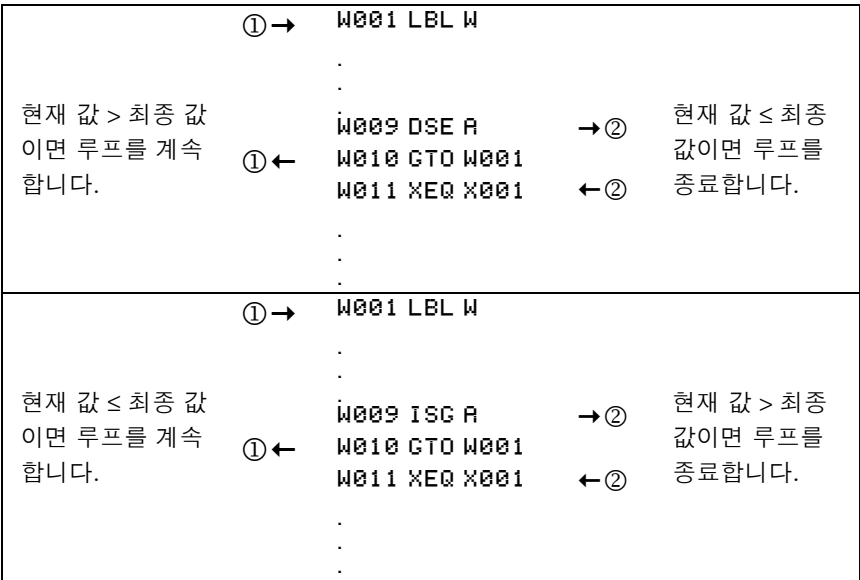
지정한 변수에 루프 제어 번호  $\pm\text{cccccc}.\text{fff}$ 가 포함되어야 합니다. 여기서 각 항목은 다음을 나타냅니다.

- $\pm\text{cccccc}$ 는 현재 카운터 값입니다(1-12자릿수). 이 값은 루프 실행과 함께 변경됩니다.
- $\text{fff}$ 는 최종 카운터 값입니다(3자릿수여야 함). 이 값은 루프 실행 시 변경되지 않습니다.  $\text{fff}$  값을 지정하지 않으면 000으로 가정합니다.

- *i*는 증가 및 감소 간격입니다(2자리숫자 또는 지정되지 않음이어야 함). 이 값은 변경되지 않습니다. *ii* 값을 지정하지 않으면 01(1씩 증가/감소)로 가정합니다.

루프 제어 번호 *cccccc.fff*가 제공되면 DSE는 *cccccc*를 *cccccc - ii*로 줄이고 새로운 *cccccc*를 *fff*와 비교한 다음  $cccccc \leq fff$ 이면 프로그램 실행 시 다음 프로그램 줄을 건너뛰도록 합니다.

루프 제어 번호 *cccccc.fff*가 제공되면 ISG는 *cccccc*를 *cccccc + ii*로 늘리고 새로운 *cccccc*를 *fff*와 비교한 다음  $cccccc > fff$ 이면 프로그램 실행 시 다음 프로그램 줄을 건너뛰도록 합니다.



예를 들어, ISG의 루프 제어 번호가 0.050이면 0에서 카운트가 시작되고 최대 50까지 카운트되며, 루프당 1씩 번호가 증가합니다.

루프 제어 번호가 복소수이거나 벡터인 경우 실수 부분이나 첫 번째 부분을 사용하여 루프를 제어합니다.

## 14-18 프로그래밍 기술

다음 프로그램에서는 ISG를 사용하여 RPN 모드에서 10회 반복합니다. 루프 카운터(1.010)는 변수 Z에 저장됩니다. 선행 및 후행 0을 삭제할 수 있습니다.

```
L001 LBL L
L002 1.01
L003 ST0 Z
L004 ISG Z
L005 GTO L004
L006 RTN
```

**[XEQ]** **[L]** **[ENTER]** 를 누른 다음 **[↵]** **[VIEW]** **[Z]** 를 눌러 이제 루프 제어 번호가 11.0100인 것을 확인합니다.

---

## 변수 및 레이블에 간접 주소 지정

간접 주소 지정은 미리 정확하게 지정하지 않고 변수 또는 레이블을 지정하기 위해 고급 프로그래밍에서 사용되는 기술입니다. 해당 항목은 프로그램 실행 시 결정되므로 프로그램의 중간 결과(또는 입력)에 따라 달라집니다.

간접 주소 지정에서는 **[I]**, **[@]**, **[J]** 및 **[@]**의 4개 키를 사용합니다.

이러한 키는 A-Z를 변수나 레이블로 사용하는 많은 함수에 대해 활성화됩니다.

- **I** 및 **J**는 내용에서 다른 변수를 참조할 수 있는 변수입니다. 다른 변수(A-Z)와 마찬가지로 숫자가 포함됩니다.
- **(I)** 및 **(J)**는 “I 또는 J의 숫자를 사용하여 주소를 지정할 변수나 레이블 확인”을 명령하는 프로그래밍 함수입니다.  
이것은 간접 주소이고, A-Z는 직접 주소입니다.

**[I]**와 **[@]**는 함께 사용되어 간접 주소를 만들며, **[J]**와 **[@]**의 경우도 마찬가지입니다.

(I) 또는 (J) 그 자체는 정의되지 않거나((I) 또는 (J)에 숫자 없음) 제어되지 않습니다 (I 또는 J에 있는 숫자 사용).

## 변수 "I" 및 "J"

다른 변수의 내용과 마찬가지로 I 또는 J의 내용을 저장, 다시 호출 및 조작할 수 있습니다. I, J의 값을 찾고 I 또는 J를 사용하여 적분할 수도 있습니다. 아래에 나열된 함수는 변수 "I"를 사용할 수 있으며, 변수 J도 동일합니다.

STO I	INPUT I	DSE I
RCL I	VIEW I	ISG I
STO +, -, ×, ÷ I	f FN d I	x < > I
RCL +, -, ×, ÷ I	SOLVE I	

## 간접 주소 (I) 및 (J)

A-Z를 변수나 레이블로 사용하는 대부분의 함수는 (I) 또는 (J)를 사용하여 A-Z(변수나 레이블) 또는 통계 레지스터를 *간접적으로* 참조할 수 있습니다. 함수 (I) 또는 (J)는 변수 I부터 J까지의 값을 사용하여 주소를 지정할 변수, 레이블 또는 레지스터를 확인합니다. 아래 표에서 그 방법을 보여 줍니다.

I/J에 포함된 값:	(I)/(J)의 주소 지정 대상:
-1	변수 A 또는 레이블 A
.	.
.	.
.	.
-26	변수 Z 또는 레이블 Z
-27	n 레지스터
-28	$\Sigma x$ 레지스터
-29	$\Sigma y$ 레지스터
-30	$\Sigma x^2$ 레지스터
-31	$\Sigma y^2$ 레지스터
-32	$\Sigma xy$ 레지스터
0	명명되지 않은 간접 변수 시작
.	.
.	.
.	.
800	최대 주소는 800입니다.
I < -32 또는 I > 800 또는 변수가 정의되지 않음	오류: INVALID (I)
J < -32 또는 J > 800 또는 변수가 정의되지 않음	오류: INVALID (J)

INPUT(I) ,INPUT(J) 및 VIEW(I) ,VIEW(J)작업은 간접 주소가 지정된 변수나 레지스터의 이름으로 디스플레이에 레이블을 지정합니다.

SUMS 메뉴를 사용하여 통계 레지스터에서 값을 다시 호출할 수 있습니다. 그러나 STO, VIEW 및 INPUT과 같은 기타 작업을 수행하려면 간접 주소 지정을 사용해야 합니다.

아래에 나열된 함수는 (I) 또는 (J)를 주소로 사용할 수 있습니다. FN=에서는 (I) 또는 (J)가 레이블을 나타내고, 다른 모든 함수에서는 (I) 또는 (J)가 변수나 레지스터를 나타냅니다.

STO(I)/(J)	INPUT(I)/(J)
RCL(I)/(J)	VIEW(I)/(J)
STO +, -,× ,÷, (I)/(J)	DSE(I)/(J)
RCL +, -,× ,÷, (I)/(J)	ISG(I)/(J)
X<>(I)/(J)	SOLVE(I)/(J)
FN=(I)/(J)	∫ FN d(I)/(J)

명명되지 않은 변수나 통계 레지스터의 값을 구하거나 적분할 수는 없습니다.

## (I)/(J)를 사용한 프로그램 제어

프로그램이 실행될 때마다 또는 동일한 프로그램의 다른 부분에서도 의 내용이 변경될 수 있으므로 STO (I) 또는 (J)와 같은 프로그램 명령은 경우에 따라 다른 변수에 값을 저장할 수 있습니다. 예를 들어, STO(-1)은 값을 변수 A에 저장함을 나타냅니다. 프로그램 실행 시까지 필요한 변수나 프로그램 레이블을 정확하게 지정하지 않고 미정 상태로 둬으로써 유연성이 유지됩니다.

간접 주소 지정은 루프 카운트 및 제어에 유용합니다. 변수 I 또는 J는 색인으로 사용되며, DSE 및 ISG 함수의 루프 제어 번호가 저장된 변수의 주소를 포함합니다.

## (I)/(J)를 사용한 방정식

방정식에 (I) 또는 (J)를 사용하여 간접적으로 변수를 지정할 수 있습니다. <I> 또는 <J>는 변수 I 또는 J의 숫자로 지정된 변수를 의미하지만(간접 참조), I 또는 J와 <I> 또는 <J>(여기서는 (I) 또는 (J) 키 대신 사용자 괄호가 사용됨)는 변수 I 또는 J를 의미합니다.

## 명명되지 않은 간접 변수

변수 I 또는 J에 양수를 넣으면 최대 801개의 간접 변수에 액세스할 수 있습니다. 다음 예에서는 간접 변수 사용 방법을 보여 줍니다.

### 프로그램 줄: (RPN 모드)

### 설명:

```
A001 LBL R
A002 100
A003 STO I
A004 12345
A005 STO (I)
```

저장 장치 주소 범위 "0-100"을 정의하고 "12345"를 주소 100에 저장했습니다.

```
A006 150
A007 STO I
A008 67890
A009 STO (I)
```

"67890"을 주소 150에 저장합니다. 이제 정의된 간접 저장 장치 범위가 "0-150"입니다.

```
A010 100
A011 STO I
A012 0
A013 STO (I)
```

0을 간접 레지스터 100에 저장합니다. 정의된 범위는 여전히 "0-150"입니다.

```
A014 170
A015 STO I
A016 RCL(I)
```

주소 "170"이 정의되지 않았으므로 "INVALID (I)"를 표시합니다.

```
A017 RTN
```



참고:

1. 정의되지 않은 저장 장치 주소에서 값을 다시 호출하려고 하면 “INVALID <I>”라는 오류 메시지가 표시됩니다. (A014 참조)
2. 계산기는 변수 0의 메모리를 0이 아닌 마지막 변수에 할당합니다. 변수를 사용한 후 메모리를 해제하기 위해 변수에 0을 저장하는 것이 중요합니다. 할당된 각 간접 레지스터는 37바이트의 프로그램 메모리를 사용합니다.
3. 최대 800개의 변수가 있습니다.





## 프로그램 값 구하기 및 적분


### 프로그램 값 구하기

7장에서는 방정식을 입력(방정식 목록에 추가됨)한 다음 변수의 값을 구하는 방법을 살펴보았습니다. 함수를 계산하는 *프로그램*을 입력하여 변수의 값을 구할 수도 있습니다. 이 기능은 특히 값을 구하는 중인 방정식이 특정 조건에서 변경되거나 반복 계산이 필요한 경우에 유용합니다.

#### 프로그래밍된 함수를 풀려면

1. 함수를 정의하는 프로그램을 입력합니다. 아래의 "SOLVE 프로그램을 작성하려면"을 참조하십시오.
2. 풀려는 프로그램을 선택합니다.  **FN**= 레이블을 누릅니다. 동일한 프로그램을 다시 풀려는 경우 이 단계를 건너뛰어도 됩니다.
3. 미지수 변수의 값을 구합니다.  **SOLVE** 변수를 누릅니다.

프로그래밍된 함수를 풀려는 경우 FN=이 필수이지만 방정식 목록의 방정식을 풀려는 경우에는 필수가 아닙니다.

계산을 중지하려면 **C** 또는 **R/S**를 누릅니다. **INTERRUPTED**라는 메시지가 줄 2에 나타납니다. 근의 현재 최적 추정값은 미지수 변수에 있습니다.  **VIEW**를 사용하여 스택을 방해하지 않고 표시합니다. 계산을 다시 시작하려면 **R/S**를 누릅니다.

#### SOLVE 프로그램을 작성하려면

이 프로그램은 방정식과 ALG 또는 RPN 연산을 가장 편리한 조합으로 사용할 수 있습니다.

1. 레이블을 사용하여 프로그램을 시작합니다. 이 레이블은 SOLVE에서 계산할 함수를 식별합니다(FN= 레이블).
2. 미지수를 비롯한 각 변수에 대한 INPUT 명령을 포함합니다. INPUT 명령을 사용하면 다중 변수 함수에 있는 모든 변수의 값을 구할 수 있습니다. 미지수에 대한 INPUT은 계산기에서 무시되므로 미지수를 비롯한 모든 변수에 대해 별도의 INPUT 명령이 포함된 한 개의 프로그램만 작성하면 됩니다.

INPUT 명령을 포함하지 않으면 프로그램에서 변수에 저장된 값이나 방정식 프롬프트에 입력된 값을 사용합니다.

3. 함수를 계산하는 명령을 입력합니다.
  - 여러 줄의 RPN 또는 ALG 시퀀스로 프로그래밍된 함수는 값을 구하면 0이 되는 수식의 형태를 사용해야 합니다. 방정식이  $f(x) = g(x)$ 이면 프로그램에서  $f(x) - g(x)$ 를 계산해야 합니다. "=0"이 암시됩니다.
  - 방정식으로 프로그래밍된 함수는 등식, 할당 또는 수식과 같은 모든 유형의 방정식이 될 수 있습니다. 방정식은 프로그램에 의해 계산되고, 값을 구하면 해당 값이 0이 됩니다. INPUT 명령을 포함하는 대신 방정식에서 변수 값을 묻는 프롬프트를 표시하려면 플래그 11이 설정되었는지 확인합니다.
4. RTN을 사용하여 프로그램을 끝냅니다. 프로그램 실행이 X 레지스터의 함수 값으로 끝나야 합니다.

### 예: ALG를 사용한 프로그램

"이상 기체 법칙" 방정식에서 미지수의 값을 구하는 ALG 연산을 사용하여 프로그램을 작성합니다. 방정식은 다음과 같습니다.

$$P \times V = N \times R \times T$$

여기서

P = 압력(기압 또는 N/m<sup>2</sup>)

V = 부피(리터)

N = 기체 몰 수

R = 보편 기체 상수

(0.0821 리터-기압/몰-K 또는 8.314 J/몰-K)

T = 온도(켈빈, K = °C + 273.1)

## 15-2 프로그램 값 구하기 및 적분

먼저 계산기를 프로그램 모드로 설정합니다. 필요한 경우 프로그램 포인터를 프로그램 메모리의 맨 위에 놓습니다.

키:	디스플레이:	설명:
<b>(ALG 모드)</b> [▶] [PRGM]		프로그램 모드를 설정합니다.
[GTO] [◦] [◦]	PRGM TOP	

프로그램을 입력합니다.

프로그램 줄: (ALG 모드)	설명:
G001 LBL G	프로그래밍된 함수를 식별합니다.
G002 INPUT P	압력을 나타내는 $P$ 를 저장합니다.
G003 INPUT V	부피를 나타내는 $V$ 를 저장합니다.
G004 INPUT N	기체 몰 수를 나타내는 $N$ 을 저장합니다.
G005 INPUT R	기체 상수를 나타내는 $R$ 을 저장합니다.
G006 INPUT T	온도를 나타내는 $T$ 을 저장합니다.
G007 $P \times V = N \times R \times T$	[EQN]을 누릅니다. 압력 $\times$ 부피 = 몰 $\times$ 기체 상수 $\times$ 온도
G008 RTN	프로그램을 끝냅니다.

체크섬 및 길이: F425 33

[C]를 눌러 프로그램 입력 모드를 취소합니다.

프로그램 "G"를 사용하여 24°C에서 2리터 병에 든 이산화탄소 0.005몰의 압력을 구합니다.

키:	디스플레이:	설명:
<b>(ALG 모드)</b> [◀] [FN] [G]		프로그램 "G"를 선택합니다.
[▶] [SOLVE] [P]	V?	SOLVE를 통해 미지수 변수 값을 계산합니다. $P$ 를 선택하고 $V$ 를 묻는 프롬프트를 표시합니다.
[2] [R/S]	값 N?	2를 $V$ 에 저장하고 $N$ 을 묻는 프롬프트를 표시합니다.
[◦] [0] [0] [5] [R/S]	값 R?	.005를 $N$ 에 저장하고 $R$ 을 묻는 프롬프트를 표시합니다.
	값	

[.] [0] [8] [2] [1]  
 [R/S]  
 [2] [4] [+][2] [7]  
 [3] [.] [1] [ENTER]  
 [R/S]

T?  
 297.1000  
 SOLVING  
 P=  
 0.0610

.0821을 R에 저장하고 T를 묻는 프롬프트를 표시합니다. T를 계산합니다.  
 297.1을 T에 저장하고 P의 값을 구합니다. 압력은 0.0610기압입니다.

### 예: 방정식을 사용한 프로그램

방정식을 사용하여 "이상 기체 법칙"을 푸는 프로그램을 작성합니다.

키:	디스플레이:	설명:
(RPN 모드)		
[>] [PRGM]	PRGM TOP	프로그램 입력 모드를 선택합니다. 프로그램 포인터를 프로그램 목록의 맨 위로 이동합니다.
[GTO] [.] [.]		프로그램에 레이블을 지정합니다. 방정식 프롬프트를 활성화합니다.
[>] [LBL] H	H001 LBL H	
[>] [FLAGS] [1]	H002 SF 11	방정식을 계산하고 플래그 11을 지웁니다(체크섬 및 길이: EDC8 9).
(1SF) [.] [1]	H003 P×V=N×R×T	
[EQN]		
[RCL] [P] [X]		
[RCL] [V] [←] [=]		
[RCL] [N] [X]		
[RCL] [R] [X]		
[RCL] [T] [ENTER]		
[>] [RTN]	H004 RTN	프로그램을 끝냅니다.
[C]	0.0610	프로그램 입력 모드를 취소합니다.
체크섬 및 프로그램 길이: DF52 21		

이제 온도가 이전 예에서 10° 떨어진 경우의 이산화탄소 압력 변화를 계산합니다.

## 15-4 프로그램 값 구하기 및 적분

키:	디스플레이:	설명:
(RPN 모드)		
$\rightarrow$ STO $\downarrow$ L	0.0610	이전 압력을 저장합니다.
$\leftarrow$ FN= H	0.0610	프로그램 "H"를 선택합니다.
$\rightarrow$ SOLVE P	V?	변수 P를 선택하고 V를 묻는 프롬프트를 표시합니다.
	2.0000	
R/S	N?	2를 V에 유지하고 N을 묻는 프롬프트를 표시합니다.
	0.0050	
R/S	R?	.005를 N에 유지하고 R을 묻는 프롬프트를 표시합니다.
	0.0821	
R/S	T?	.0821을 R에 유지하고 T를 묻는 프롬프트를 표시합니다.
	297.1000	
ENTER 1 0 =	T?	새로운 T를 계산합니다.
	287.1000	
R/S	SOLVING	287.1을 T에 저장하고 새로운 P의 값을 구합니다.
	P=	
	0.0589	
RCL L =	-0.0021	온도가 297.1K에서 287.1K로 떨어진 경우의 기체 압력 변화를 계산합니다(결과가 음수로 나오면 압력 저하를 나타냄).

---

## 프로그램에 SOLVE 사용

프로그램의 일부로 SOLVE 연산을 사용할 수 있습니다.

해당되는 경우 SOLVE 변수 명령을 사용하기 전에 초기 추정값을 미지수 변수와 X 레지스터에 포함하거나 프롬프트를 표시합니다. 방정식에서 미지수 변수의 값을 구하는 두 명령은 프로그램에 다음과 같이 나타납니다.

FN= 레이블

SOLVE 변수

프로그래밍된 SOLVE 명령은 프로그램에 의미 있는 출력이 아닐 수도 있으므로(즉, 이 숫자를 표시하기 전에 이 숫자로 추가 계산을 수행할 수도 있으므로) 레이블이 지정된 디스플레이(변수 = 값)를 생성하지 않습니다. 이 결과를 표시하려는 경우 VIEW 변수 명령을 SOLVE 명령 뒤에 추가합니다.

미지수 변수에 대한 값을 구할 수 없는 경우 14장에서 설명한 "참이면 수행" 규칙에 따라 다음 프로그램 줄을 건너뛵니다. 그런 후에 프로그램에서 새 초기 추정값을 선택하거나 입력 값을 변경하여 근을 구하지 못한 경우를 처리해야 합니다.

### 예: 프로그램에서의 SOLVE

다음 내용은 **[XEQ]** X 또는 Y를 눌러  $x$  또는  $y$ 의 값을 구할 수 있는 프로그램에서 추출한 것입니다.



**프로그램 줄:**  
**(RPN 모드)**

X001 LBL X  
X002 24  
X003 GTO L001  
체크섬 및 길이: 62A0 11  
Y001 LBL Y  
Y002 25  
Y003 GTO L001  
체크섬 및 길이: 221E 11  
L001 LBL L  
L002 STO I  
L003 FN= F  
L004 SOLVE(I)  
L005 VIEW(I)  
L006 RTN  
체크섬 및 길이: D45B 18  
F001 LBL F  
. .  
F010 RTN

**설명:**

X의 설정입니다.  
X의 색인입니다.  
주 루틴으로 분기합니다.

Y의 설정입니다.  
Y의 색인입니다.  
주 루틴으로 분기합니다.

주 루틴입니다.  
색인을 I에 저장합니다.  
풀려는 프로그램을 정의합니다.  
해당 변수의 값을 구합니다.  
해를 표시합니다.  
프로그램을 끝냅니다.

$f(x,y)$ 를 계산합니다. 필요에 따라 INPUT 또는 방정식 프롬프트를 포함합니다.

## 프로그램 적분

8장에서는 방정식 목록에 추가된 방정식 또는 수식을 입력한 다음 변수와 관련해서 적분할 수 있는 방법을 살펴보았습니다. 함수를 계산하는 프로그램을 입력한 다음 변수와 관련해서 적분할 수도 있습니다. 이 기능은 특히 적분 중인 함수가 특정 조건에서 변경되거나 반복 계산이 필요한 경우에 유용합니다.

### 프로그래밍된 함수를 적분하려면

1. 피적분 함수를 정의하는 프로그램을 입력합니다. 아래의 "[ FN 프로그램을 작성하려면]"을 참조하십시오.
2. 적분할 함수를 정의하는 프로그램을 선택합니다. **[↵] [FN=]** 레이블을 누릅니다. 동일한 프로그램을 다시 적분하는 경우 이 단계를 건너뛰어도 됩니다.
3. 적분 한도를 입력합니다. *하한값*을 입력하고 **[ENTER]**를 누른 다음 *상한값*을 입력합니다.
4. 적분 변수를 선택하고 계산을 시작합니다. **[↵] [↵]** 변수를 누릅니다.

프로그래밍된 함수를 적분하는 경우 FN=이 필수이지만 방정식 목록의 방정식을 적분하는 경우에는 필수가 아닙니다.

**[C]** 또는 **[R/S]**를 눌러 실행 중인 적분 계산을 중지할 수 있으며, **INTERRUPTED**라는 메시지가 줄 2에 나타납니다. 그러나 계산을 다시 시작할 수는 없습니다. 계산이 정상적으로 완료될 때까지 적분에 대한 정보를 사용할 수 없습니다.

적분 계산이 실행되는 동안 **[XEQ]**를 누르면 **[↵] [FN=]** 연산이 취소됩니다. 이 경우 처음부터 다시 **[↵] [FN=]**을 시작해야 합니다.

### [ FN 프로그램을 작성하려면

이 프로그램은 방정식, ALG 또는 RPN 연산을 가장 편리한 조합으로 사용할 수 있습니다.

1. 레이블을 사용하여 프로그램을 시작합니다. 이 레이블은 적분할 함수를 식별합니다(FN= 레이블).

- 적분 변수를 비롯한 각 변수에 대한 INPUT 명령을 포함합니다. INPUT 명령을 사용하면 다중 변수 함수에 있는 모든 변수와 관련해서 적분할 수 있습니다. 적분 변수에 대한 INPUT은 계산기에서 무시되므로 적분 변수를 비롯한 모든 변수에 대해 별도의 INPUT 명령이 포함된 한 개의 프로그램만 작성하면 됩니다.

INPUT 명령을 포함하지 않으면 프로그램에서 변수에 저장된 값이나 방정식 프롬프트에 입력된 값을 사용합니다.

- 함수를 계산하는 명령을 입력합니다.
  - 여러 줄의 RPN 또는 ALG 시퀀스로 프로그래밍된 함수에서 적분하려는 함수 값을 계산해야 합니다.
  - 방정식으로 프로그래밍된 함수는 모든 유형의 방정식이 될 수 있지만 대체로 피적분 함수를 지정하는 수식으로 포함됩니다. INPUT 명령을 포함하는 대신 방정식에서 변수 값을 묻는 프롬프트를 표시하려면 플래그 11이 설정되었는지 확인합니다.
- RTN을 사용하여 프로그램을 끝냅니다. 프로그램 실행이 X 레지스터의 함수 값으로 끝나야 합니다.

### 예: 방정식을 사용한 프로그램

8장의 예에서 사용된 사인 적분 함수는 다음과 같습니다.

$$Si(t) = \int_0^t \left( \frac{\sin x}{x} \right) dx$$

피적분 함수를 정의하는 프로그램을 적분하여 이 함수를 계산할 수 있습니다.

```
S001 LBL S           함수를 정의합니다.
S002 SIN(X)÷X       수식으로서의 함수입니다. (체크섬 및 길이: 0EE0 8)
S003 RTN           서브루틴을 끝냅니다.
체크섬 및 프로그램 길이: D57E 17
```

이 프로그램을 입력하고  $x$ 와 관련해서 0부터 2( $t=2$ )까지 사인 적분 함수를 적분합니다.

키:	디스플레이:	설명:
(RPN 모드) [MODE] [2] (2RAD)		
[↵] [FN=] [S]		라디안 모드를 선택합니다.
[0] [ENTER] [2]	2_	레이블 S를 피적분 함수로 선택합니다.
[↵] [∫] [X]	INTEGRATING ∫ =	적분의 하한값과 상한값을 입력합니다.
[MODE] [1] (1DEG)	1.6054 1.6054	0부터 2까지 함수를 적분하고 결과를 표시합니다.
		도 모드로 복원합니다.

## 프로그램에 적분 사용

프로그램에서 적분을 실행할 수 있습니다. 적분을 실행하기 전에 적분 한도를 포함하거나 프롬프트를 표시해야 하며, 정확성과 실행 시간은 프로그램 실행 시의 디스플레이 형식에 의해 제어됨을 명심하십시오. 두 개의 적분 명령은 프로그램에 다음과 같이 나타납니다.

FN= 레이블

∫FN d 변수

*프로그래밍된* ∫FN 명령은 프로그램에 의미 있는 출력이 아닐 수도 있으므로(즉, 이 숫자를 표시하기 전에 이 숫자로 추가 계산을 수행할 수도 있으므로) 레이블이 지정된 디스플레이(∫ =    )를 생성하지 않습니다. 이 결과를 *표시하려는* 경우 PSE ([↵] [PSE]) 또는 STOP([R/S]) 명령을 추가하여 X 레지스터의 ∫FN 명령 뒤에 결과를 표시합니다.

각 적분 또는 값 구하기를 반복하는 동안 표시되는 방정식(플래그 10 설정) 바로 뒤에 PSE 명령이 오는 경우 방정식이 1초 동안 표시되고 각 반복이 끝날 때까지 실행이 계속됩니다. 방정식이 표시되는 동안에는 스크롤 또는 키보드 입력을 사용할 수 없습니다.

## 15-10 프로그램 값 구하기 및 적분

## 예: 프로그램의 ∫ FN

16장의 "정규 분포 및 역 정규 분포" 프로그램에는 아래와 같은 완전 밀도 함수 방정식의 적분이 포함되어 있습니다.

$$\frac{1}{S\sqrt{2\pi}} \int_M^D e^{-\left(\frac{D-M}{S}\right)^2/2} dD.$$

$e^{-(D-M)^2/S^2}$  함수는 F 레이블이 지정된 루틴에서 계산됩니다. 다른 루틴에서는 알려진 값을 묻는 프롬프트를 표시하고 기타 계산을 수행하여 정규 곡선의 위쪽 꼬리 영역인  $Q(D)$ 를 구합니다. 적분 자체는 루틴 Q에서 설정 및 실행됩니다.

Q001 LBL Q

Q002 RCL M

Q003 RCL X

Q004 FN= F

Q005 ∫ FN d D

적분의 하한값을 다시 호출합니다.

적분의 상한값을 다시 호출합니다. ( $X = D$ )

함수를 지정합니다.

더미 변수  $D$ 를 사용하여 정규 함수를 적분합니다.

---

## 값 구하기와 적분의 한계

SOLVE 변수 및 ∫ FN d 변수 명령은 다른 SOLVE 또는 ∫ FN 명령이 포함된 루틴을 호출할 수 없습니다. 즉, 두 함수는 모두 재귀적으로 사용할 수 없습니다. 예를 들어, 중적분을 계산하려고 하면 ∫ ∫ FN 오류가 발생합니다. 또한 SOLVE 및 ∫ FN은 FN= 레이블 명령이 포함된 루틴을 호출할 수 없습니다. 호출하려고 하면 SOLVE ACTIVE 또는 ∫ FN ACTIVE 오류가 반환됩니다. ∫ FN에서 SOLVE 명령이 포함된 루틴을 호출할 수 없는 것과 마찬가지로(∫ ∫ SOLVE) 오류 발생), SOLVE는 ∫ FN 명령이 포함된 루틴을 호출할 수 없습니다(SOLVE ∫ ∫ FN) 오류 발생).

프로그램의 SOLVE 변수 및 ∫ FN d 변수 명령은 계산기에 보류 중인 20개 서브루틴 리턴 중 하나를 사용합니다. 14장의 "중첩된 서브루틴"을 참조하십시오.



## 통계 프로그램

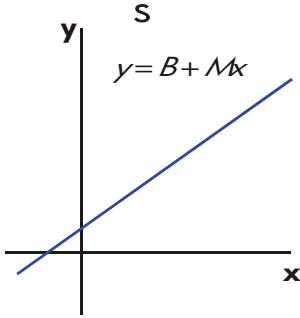
---

### 곡선 맞춤

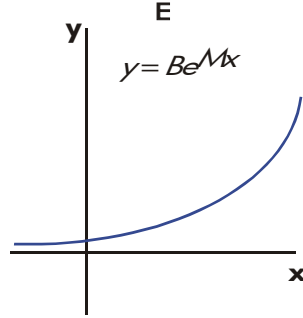
이 프로그램은 네 가지 방정식 모델 중 하나를 데이터에 맞추는 데 사용할 수 있습니다. 네 모델은 직선, 로그 곡선, 지수 곡선 및 거듭제곱 곡선입니다. 이 프로그램은 둘 이상의  $(x, y)$  데이터 쌍을 받아들인 다음 상관계수  $r$  및 두 개의 회귀계수  $m$  과  $b$ 를 계산합니다. 이 프로그램에는 추정값  $\hat{x}$ 와  $\hat{y}$ 를 계산하는 루틴이 포함되어 있습니다. 이러한 값의 정의는 12장의 "선형 회귀"를 참조하십시오.

아래에는 곡선 및 관련 방정식의 샘플이 나와 있습니다. HP 35s의 내부 회귀 함수를 사용하여 회귀계수를 계산합니다.

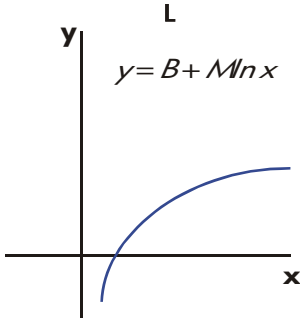
Straight Line Fit



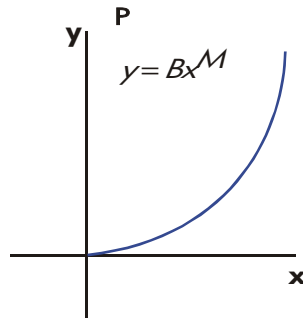
Exponential Curve Fit



Logarithmic Curve Fit



Power Curve Fit





로그 곡선을 맞추려면  $x$  값이 양수여야 합니다. 지수 곡선을 맞추려면  $y$  값이 양수여야 합니다. 거듭제곱 곡선을 맞추려면  $x$ 와  $y$ 가 모두 양수여야 합니다. 이러한 경우에 음수를 입력하면 LOG(NEG) 오류가 발생합니다.

데이터 값의 크기가 크지만 차이가 비교적 적은 데이터 값은 크기 차이가 큰 데이터 값과 마찬가지로 정밀도 문제를 발생시킬 수 있습니다. 12장의 "데이터 정밀도의 한계"를 참조하십시오.



## 프로그램 목록:

프로그램 줄: (RPN 모드)	설명
S001 LBL S	이 루틴은 직선 모델의 상태를 설정합니다.
S002 CF 0	X에서의 표시기인 플래그 0을 지웁니다.
S003 CF 1	Y의 표시기인 플래그 1을 지웁니다.
S004 GTO Z001	공동 입력 지점 Z로 분기합니다.
체크섬 및 길이: 8E85 12	
L001 LBL L	이 루틴은 로그 모델의 상태를 설정합니다.
L002 SF 0	X에서의 표시기인 플래그 0을 설정합니다.
L003 CF 1	Y에서의 표시기인 플래그 1을 지웁니다.
L004 GTO Z001	공동 입력 지점 Z로 분기합니다.
체크섬 및 길이: AD1B 12	
E001 LBL E	이 루틴은 지수 모델의 상태를 설정합니다.
E002 CF 0	X에서의 표시기인 플래그 0을 지웁니다.
E003 SF 1	Y에서의 표시기인 플래그 1을 설정합니다.
E004 GTO Z001	공동 입력 지점 Z로 분기합니다.
체크섬 및 길이: D6F1 12	
P001 LBL P	이 루틴은 거듭제곱 모델의 상태를 설정합니다.
P002 SF 0	X에서의 표시기인 플래그 0을 설정합니다.
P003 SF 1	Y에서의 표시기인 플래그 1을 설정합니다.
체크섬 및 길이: 3800 9	
Z001 LBL Z	모든 모델의 공동 입력 지점을 정의합니다.
Z002 CLΣ	통계 레지스터를 지웁니다.  CLEAR  (4Σ)를 누릅니다.
Z003 0	첫 번째 입력에 대해 루프 카운터를 0으로 설정합니다.
체크섬 및 길이: 8611 10	
W001 LBL W	입력 루프의 시작을 정의합니다.
W002 1	루프 카운터를 1만큼 조정하여 입력을 묻는 프롬프트를 표시합니다.
W003 +	
W004 STO X	X를 묻는 프롬프트와 함께 표시되도록 루프 카운터를 X에 저장합니다.
W005 INPUT X	프롬프트와 함께 카운터를 표시하고 X입력을 저장합니다.

**프로그램 줄:**  
**(RPN 모드)**

**설명**

W006 FS? 0      플래그 0을 설정한 경우 ...  
W007 LN      ... X 입력의 자연 로그를 수행합니다.  
W008 STO B      수정 루틴에 대해 해당 값을 저장합니다.  
W009 INPUT Y      Y를 묻는 프롬프트를 표시하고 저장합니다.  
W010 FS? 1      플래그 1을 설정한 경우 ...  
W011 LN      ... Y 입력의 자연 로그를 수행합니다.  
W012 STO R  
W013 RCL B  
W014  $\Sigma+$       B와 R을 통계 레지스터에  $x, y$  데이터 쌍으로 누적합니다.  
W015 GTO W001      다른 X, Y 쌍에 대해 반복합니다.  
    체크섬 및 길이: 9560 46

U001 LBL U      "실행 취소" 루틴의 시작을 정의합니다.  
U002 RCL R      가장 최근 데이터 쌍을 다시 호출합니다.  
U003 RCL B  
U004  $\Sigma-$       통계 누적에서 이 쌍을 삭제합니다.  
U005 GTO W001      다른 X, Y 쌍에 대해 반복합니다.  
    체크섬 및 길이: A79F 15

R001 LBL R      출력 루틴의 시작을 정의합니다.  
R002 r      상관계수를 계산합니다.  
R003 STO R      R에 저장합니다.  
R004 VIEW R      상관계수를 표시합니다.  
R005 b      계수 b를 계산합니다.  
R006 FS? 1      플래그 1을 설정하면 b의 자연 역대수를 수행합니다.  
R007 eX  
R008 STO B      b를 B에 저장합니다.  
R009 VIEW B      값을 표시합니다.  
R010 m      계수 m을 계산합니다.  
R011 STO M      m을 M에 저장합니다.  
R012 VIEW M      값을 표시합니다.  
    체크섬 및 길이: 850C 36

Y001 LBL Y      추산(예측) 루프의 시작을 정의합니다.  
Y002 INPUT X      x 값을 표시하고, 프롬프트를 표시하고, 변경된 경우 X에  
    저장합니다.  
Y003 FS?0      플래그 0을 설정한 경우 ...

**프로그램 줄:  
(RPN 모드)**

**설명**

Y004 GTO K001 K001로 분기합니다.  
Y005 GTO M001 M001로 분기합니다.  
Y006 STO Y  $\hat{y}$  값을 Y에 저장합니다.  
Y007 INPUT Y y 값을 표시하고, 프롬프트를 표시하고, 변경된 경우 Y에 저장합니다.  
Y008 FS?0 플래그 0을 설정한 경우 ...  
Y009 GTO O001 O001로 분기합니다.  
Y010 GTO N001 N001로 분기합니다.  
Y011 STO X 다음 루프에 대해  $\hat{x}$ 를 X에 저장합니다.  
Y012 GTO Y001 다른 추정값에 대해 반복합니다.  
체크섬 및 길이: C3B7 36

A001 LBL A 이 서브루틴에서는 직선 모델의  $\hat{y}$ 를 계산합니다.  
A002 RCL M  
A003 RCL× X  
A004 RCL+ B  $\hat{y} = MX + B$ 를 계산합니다.  
A005 RTN 호출 루틴으로 돌아갑니다.  
체크섬 및 길이: 9688 15

G001 LBL G 이 서브루틴에서는 직선 모델의  $\hat{x}$ 를 계산합니다.  
G002 RCL Y  
G003 RCL- B  
G004 RCL÷ M  $\hat{x} = (Y - B) ÷ M$ 를 계산합니다.  
G005 RTN 호출 루틴으로 돌아갑니다.  
체크섬 및 길이: 9C0F 15

B001 LBL B 이 서브루틴에서는 로그 모델의  $\hat{y}$ 를 계산합니다.  
B002 RCL X  
B003 LN  
B004 RCL× M  
B005 RCL+ B  $\hat{y} = M \ln X + B$ 를 계산합니다.  
B006 RTN 호출 루틴으로 돌아갑니다.  
체크섬 및 길이: 889C 18

H001 LBL H 이 서브루틴에서는 로그 모델의  $\hat{x}$ 를 계산합니다.  
H002 RCL Y

**프로그램 줄:  
(RPN 모드)**

**설명**

H003 RCL - B

H004 RCL ÷ M

H005 e<sup>x</sup>                     $\hat{x} = e(Y - B) \div M$ 을 계산합니다.

H006 RTN                호출 루틴으로 돌아갑니다.

체크섬 및 길이: ODBE 18

C001 LBL C             이 서브루틴에서는 지수 모델의  $\hat{y}$ 를 계산합니다.

C002 RCL M

C003 RCL × X

C004 e<sup>x</sup>

C005 RCL × B            $\hat{y} = Be^{MX}$ 를 계산합니다.

C006 GTO M005        M005로 분기합니다.

체크섬 및 길이: 9327 18

I001 LBL I             이 서브루틴에서는 지수 모델의  $\hat{x}$ 를 계산합니다.

I002 RCL Y

I003 RCL ÷ B

I004 LN

I005 RCL ÷ M            $\hat{x} = (\ln(Y - B)) \div M$ 을 계산합니다.

I006 GTO N005        N005로 이동합니다.

체크섬 및 길이: 7219 18

D001 LBL D             이 서브루틴에서는 거듭제곱 모델의  $\hat{y}$ 를 계산합니다.

D002 RCL X

D003 RCL M

D004 y<sup>x</sup>

D005 RCL × B            $Y = B(X^M)$ 을 계산합니다.

D006 GTO K005        K005로 이동합니다.

체크섬 및 길이: 11B3 18

J001 LBL J             이 서브루틴에서는 거듭제곱 모델의  $\hat{x}$ 를 계산합니다.

J002 RCL Y

J003 RCL ÷ B

J004 RCL M

J005 1/x

J006 y<sup>x</sup>                 $\hat{x} = (Y/B)^{1/M}$ 을 계산합니다.

J007GTO O005        O005로 이동합니다.

**프로그램 줄:  
(RPN 모드)**

**설명**

체크섬 및 길이: 8524 21

K001 LBL K	D001 또는 B001의 실행 여부를 결정합니다.
K002 FS?1	플래그 1을 설정한 경우 ...
K003 XEQ D001	D001을 실행합니다.
K004 XEQ B001	B001을 실행합니다.
K005 GTO Y006	Y006으로 이동합니다.

체크섬 및 길이: 4BFA 15

M001 LBL M	C001 또는 A001의 실행 여부를 결정합니다.
M002 FS?1	플래그 1을 설정한 경우 ...
M003 XEQ C001	C001을 실행합니다.
M004 XEQ A001	A001을 실행합니다.
M005 GTO Y006	Y006으로 이동합니다.

체크섬 및 길이: 1C4D 15

0001 LBL O	J001 또는 H001의 실행 여부를 결정합니다.
0002 FS?1	플래그 1을 설정한 경우 ...
0003 XEQ J001	J001을 실행합니다.
0004 XEQ H001	H001을 실행합니다.
0005 GTO Y011	Y011로 이동합니다.

체크섬 및 길이: 0AA5 15

N001 LBL N	I001 또는 G001의 실행 여부를 결정합니다.
N002 FS?1	플래그 1을 설정한 경우 ...
N003 XEQ I001	I001을 실행합니다.
N004 XEQ G001	G001을 실행합니다.
N005 GTO Y011	Y011로 이동합니다.

체크섬 및 길이: 666D 15

## 사용되는 플래그:

X입력에 대해 자연 로그가 필요한 경우 플래그 0이 설정됩니다. Y입력에 대해 자연 로그가 필요한 경우 플래그 1이 설정됩니다.

루틴 N에 플래그 1을 설정하면 I001이 실행됩니다. 플래그 1을 지우면 G001이 실행됩니다.

## 프로그램 명령:

1. 프로그램 루틴을 입력하고, 완료되면 **C**를 누릅니다.
2. **XEQ**를 누르고 다음을 눌러 맞추려는 곡선 유형을 선택합니다.
  - 직선의 경우 **S** **ENTER**
  - 로그 곡선의 경우 **L** **ENTER**
  - 지수 곡선의 경우 **E** **ENTER**
  - 거듭제곱 곡선의 경우 **P** **ENTER**
3.  $x$  값을 입력하고 **R/S**를 누릅니다.
4.  $y$  값을 입력하고 **R/S**를 누릅니다.
5. 각 데이터 쌍에 대해 3단계와 4단계를 반복합니다. 3단계에서 **R/S**를 누른 후 오류를 발견하게 되면(Y? 값 프롬프트가 여전히 표시되는 경우) **R/S**를 다시 누른 다음(X? 값 프롬프트 표시) **XEQ** **U** **ENTER**를 눌러 마지막 데이터 쌍을 실행 취소(제거)합니다. 4단계 후에 오류를 발견한 경우에는 **XEQ** **U** **ENTER**를 누릅니다. 두 경우 모두, 3단계에서 계속합니다.
6. 모든 데이터를 입력한 후 **XEQ** **R** **ENTER**를 눌러 상관계수  $R$ 를 확인합니다.
7. **R/S**를 눌러 회귀계수  $B$ 를 확인합니다.
8. **R/S**를 눌러 회귀계수  $M$ 을 확인합니다.
9. **R/S**를 눌러  $\hat{x}$ ,  $\hat{y}$  추산 루틴에 대한 X? 값 프롬프트를 확인합니다.
10.  $x$ 를 기반으로  $\hat{y}$ 를 추정하려는 경우 X? 값 프롬프트에  $x$ 를 입력한 다음 **R/S**를 눌러  $\hat{y}$  (Y?)를 확인합니다.
11.  $y$ 를 기반으로  $\hat{x}$ 를 추정하려는 경우 Y? 값 프롬프트가 표시될 때까지 **R/S**를 누르고  $y$ 를 입력한 다음 **R/S**를 눌러  $\hat{x}$  (X?)를 확인합니다.
12. 추산을 더 수행하려면 10단계나 11단계로 이동합니다.
13. 새로운 사례의 경우 2단계로 이동합니다.

**사용되는 변수:**

$B$	회귀계수(직선의 $y$ 인터셉트)이며 스크래치에도 사용됩니다.
$M$	회귀계수(직선의 기울기)입니다.
$R$	상관계수이며 스크래치에도 사용됩니다.
$X$	데이터를 입력하는 경우 데이터 쌍의 $x$ 이거나, $\hat{y}$ 를 추정하는 경우 가상 $x$ 이거나, 가상 $y$ 가 제공된 경우 $\hat{x}$ ( $x$ 추정치)입니다.
$Y$	데이터를 입력하는 경우 데이터 쌍의 $y$ 이거나, $\hat{x}$ 를 추정하는 경우 가상 $y$ 이거나, 가상 $x$ 가 제공된 경우 $\hat{y}$ ( $y$ 추정치)입니다.
통계 레지스터	통계 누적 및 계산

**예 1:**

직선을 아래 데이터에 맞춥니다. 세 번째 데이터 쌍을 입력할 때 의도적으로 오류를 만들고 실행 취소 루틴을 사용하여 수정합니다. 또한  $x$  값 37의  $y$ 를 추정하고,  $y$  값 101의  $x$ 를 추정합니다.

X	40.5	38.6	37.9	36.2	35.1	34.6
Y	104.5	102	100	97.5	95.5	94

키:	디스플레이:	설명:
(RPN 모드)		
<b>XEQ</b> <b>S</b> <b>ENTER</b>	X? 1.0000	직선 루틴을 시작합니다.
<b>4</b> <b>0</b> <b>.</b> <b>5</b> <b>R/S</b>	Y? 값 X? 2.0000	데이터 쌍의 $x$ 값을 입력합니다.
<b>1</b> <b>0</b> <b>4</b> <b>.</b> <b>5</b>	X? 2.0000	데이터 쌍의 $y$ 값을 입력합니다.
<b>R/S</b>	Y? 104.5000	데이터 쌍의 $x$ 값을 입력합니다.
<b>3</b> <b>8</b> <b>.</b> <b>6</b> <b>R/S</b>	X? 3.0000	데이터 쌍의 $y$ 값을 입력합니다.
<b>1</b> <b>0</b> <b>2</b> <b>R/S</b>		

이제 잘못된 입력을 수정하는 방법을 확인할 수 있도록 일부러 37.9 대신 379를 입력합니다.

키:	디스플레이:	설명:
(RPN 모드)		
<b>3</b> <b>7</b> <b>9</b> <b>R/S</b>	Y? 102.0000	데이터 쌍의 잘못된 $x$ 값을 입력합니다.
<b>R/S</b>	X? 4.0000	X? 프롬프트를 불러옵니다.
<b>XEQ</b> <b>U</b> <b>ENTER</b>	X? 3.0000	마지막 쌍을 삭제합니다. 이제 올바른 데이터 입력으로 계속합니다.
<b>3</b> <b>7</b> <b>.</b> <b>9</b> <b>R/S</b>	Y? 102.0000	데이터 쌍의 올바른 $x$ 값을 입력합니다.
<b>1</b> <b>0</b> <b>0</b> <b>R/S</b>	X? 4.0000	데이터 쌍의 $y$ 값을 입력합니다.
<b>3</b> <b>6</b> <b>.</b> <b>2</b> <b>R/S</b>	Y? 100.0000	데이터 쌍의 $x$ 값을 입력합니다.
<b>9</b> <b>7</b> <b>.</b> <b>5</b> <b>R/S</b>	X? 5.0000	데이터 쌍의 $y$ 값을 입력합니다.
<b>3</b> <b>5</b> <b>.</b> <b>1</b> <b>R/S</b>	Y? 97.5000	데이터 쌍의 $x$ 값을 입력합니다.
<b>9</b> <b>5</b> <b>.</b> <b>5</b> <b>R/S</b>	X? 6.0000	데이터 쌍의 $y$ 값을 입력합니다.
<b>3</b> <b>4</b> <b>.</b> <b>6</b> <b>R/S</b>	Y? 95.5000	데이터 쌍의 $x$ 값을 입력합니다.
<b>9</b> <b>4</b> <b>R/S</b>	X? 7.0000	데이터 쌍의 $y$ 값을 입력합니다.
<b>XEQ</b> <b>R</b> <b>ENTER</b>	R= 0.9955	상관계수를 계산합니다.
<b>R/S</b>	B= 33.5271	회귀계수 $B$ 를 계산합니다.
<b>R/S</b>	M= 1.7601	회귀계수 $M$ 을 계산합니다.
<b>R/S</b>	X? 7.0000	가상 $x$ 값을 묻는 프롬프트를 표시합니다.
<b>3</b> <b>7</b> <b>R/S</b>	Y? 98.6526	37을 X에 저장하고 $\hat{y}$ 를 계산합니다.
<b>1</b> <b>0</b> <b>1</b> <b>R/S</b>	X? 38.3336	101을 Y에 저장하고 $\hat{x}$ 를 계산합니다.



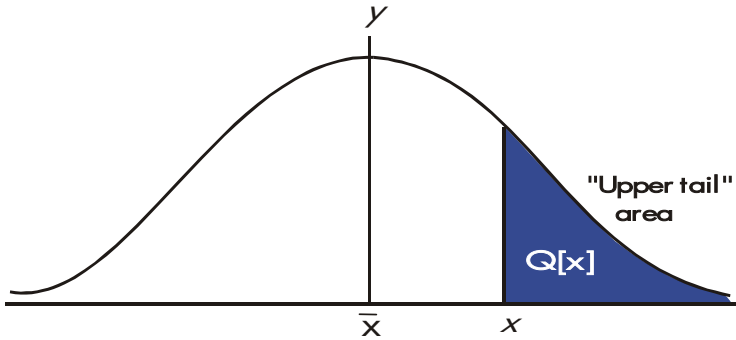
**예 2:**

동일한 데이터를 사용하여 로그, 지수 및 거듭제곱 곡선 맞춤에 대해 예 1을 반복합니다. 아래 표에서는 각 곡선 유형의 시작 실행 레이블과 결과(계수 및 회귀계수와  $x$  및  $y$  추정값)를 제공합니다. 다른 곡선 맞춤에 대해 프로그램을 실행할 때마다 데이터 값을 다시 입력해야 합니다.

	<b>로그</b>	<b>지수</b>	<b>거듭제곱</b>
시작 방법:	<b>XEQ</b> <b>[L]</b> <b>ENTER</b>	<b>XEQ</b> <b>[E]</b> <b>ENTER</b>	<b>XEQ</b> <b>[P]</b> <b>ENTER</b>
$R$	0.9965	0.9945	0.9959
$B$	-139.0088	51.1312	8.9730
$M$	65.8446	0.0177	0.6640
$Y(X=37\text{일 때의 } \hat{y})$	98.7508	98.5870	98.6845
$X(Y=101\text{일 때의 } \hat{x})$	38.2857	38.3628	38.3151

## 정규 분포 및 역 정규 분포

정규 분포는 평균을 중심으로 무작위 변화량을 모델링하는 데 자주 사용됩니다. 이 모델에서는 표본 분포가 표준 편차  $S$ 를 사용하여 평균  $M$ 을 중심으로 대칭이며 아래에 표시된 종 모양의 곡선 모양과 비슷하다고 가정합니다. 값  $x$ 가 제공되면 이 프로그램은 표본 데이터의 무작위 선택 값이 더 클 확률을 계산합니다. 이것을 위쪽 꼬리 영역  $Q(x)$ 라고 합니다. 또한 이 프로그램에서는 역함수를 제공합니다. 값  $Q(x)$ 가 제공되면 프로그램에서 해당  $x$  값을 계산합니다.



$$Q(x) = 0.5 - \frac{1}{\sigma\sqrt{2\pi}} \int_{\bar{x}}^x e^{-((x-\bar{x})+\sigma)^2+2} dx$$

이 프로그램에서는 HP 35s의 내장된 적분 기능을 사용하여 정규 도수 곡선의 방정식을 적분합니다. 역을 구하려면 제공된 확률  $Q(x)$ 를 산출하는  $x$  값을 반복해서 검색하는 뉴턴의 방법을 사용합니다.

## 프로그램 목록:

프로그램 줄: (RPN 모드)	설명
S001 LBL S	이 루틴에서는 정규 분포 프로그램을 초기화합니다.
S002 0	평균에 대해 기본값을 저장합니다.
S003 STO M	
S004 INPUT M	평균 $M$ 을 묻는 프롬프트를 표시하고 저장합니다.
S005 1	표준 편차에 대해 기본값을 저장합니다.
S006 STO S	
S007 INPUT S	표준 편차 $S$ 을 묻는 프롬프트를 표시하고 저장합니다.
S008 RTN	표준 편차 값 표시를 중지합니다.
체크섬 및 길이: 70BF 26	
D001 LBL D	$X$ 가 제공된 경우 이 루틴에서는 $Q(X)$ 를 계산합니다.
D002 INPUT X	$X$ 을 묻는 프롬프트를 표시하고 저장합니다.
D003 XEQ Q001	위쪽 꼬리 영역을 계산합니다.
D004 STO Q	VIEW 함수로 표시할 수 있도록 값을 $Q$ 에 저장합니다.
D005 VIEW Q	$Q(X)$ 를 표시합니다.
D006 GTO D001	루프를 통해 다른 $Q(X)$ 를 계산합니다.
체크섬 및 길이: 042A 18	
I001 LBL I	$Q(X)$ 가 제공된 경우 이 루틴에서는 $X$ 을 계산합니다.
I002 INPUT Q	$Q(X)$ 를 묻는 프롬프트를 표시하고 저장합니다.
I003 RCL M	평균을 다시 호출합니다.
I004 STO X	평균을 $X_{\text{guess}}$ 라는 $X$ 추정값으로 저장합니다.
체크섬 및 길이: A970 12	
T001 LBL T	이 레이블에서는 반복 루프의 시작을 정의합니다.
T002 XEQ Q001	$(Q(X_{\text{guess}}) - Q(X))$ 를 계산합니다.
T003 RCL - Q	
T004 RCL X	
T005 STO D	
T006 R↓	
T007 XEQ F001	$X_{\text{guess}}$ 에서 도함수를 계산합니다.
T008 RCL ÷ T	
T009 ÷	$X_{\text{guess}}$ 의 수정을 계산합니다.
T010 STO+ X	수정을 더하여 새로운 $X_{\text{guess}}$ 를 구합니다.
T011 ABS	
T012 0.0001	

**프로그램 줄:  
(RPN 모드)**

**설명**

T013  $\times \vee ?$  테스트를 통해 수정이 의미가 있는지 확인합니다.  
 T014 GTO T001 수정이 의미가 있는 경우 루프의 시작 부분으로 돌아갑니다.  
 수정이 의미가 없는 경우 계속합니다.  
 T015 RCL X  
 T016 VIEW X X의 계산된 값을 표시합니다.  
 T017 GTO I001 루프를 통해 다른 X를 계산합니다.  
 체크섬 및 길이: EDF4 57

Q001 LBL Q 이 서브루틴에서는 위쪽 꼬리 영역  $Q(x)$ 를 계산합니다.  
 Q002 RCL M 적분의 하한값을 다시 호출합니다.  
 Q003 RCL X 적분의 상한값을 다시 호출합니다.  
 Q004 FN= F 적분을 위해 LBL F에 의해 정의된 함수를 선택합니다.  
 Q005  $\int$  FN  $\div$  D 더미 변수 D를 사용하여 정규 함수를 적분합니다.  
 Q006 2  
 Q007  $\pi$   
 Q008  $\times$   
 Q009  $\sqrt{\times}$   
 Q010 RCL  $\times$  S  $S \times \sqrt{2\pi}$ 를 계산합니다.  
 Q011 STO T 역 루틴을 위해 결과를 일시적으로 저장합니다.  
 Q012  $\div$   
 Q013  $+/-$   
 Q014 0.5  
 Q015 + 평균을 하한값으로 사용하여 적분했기 때문에 곡선 아래  
 면적의 절반을 더합니다.  
 Q0016 RTN 호출 루틴으로 돌아갑니다.  
 체크섬 및 길이: 8387 52

F001 LBL F 이 서브루틴에서는 정규 함수  $e^{-((X-M)+S)^2+2}$ 의 피적분  
 함수를 계산합니다.

F002 RCL D  
 F003 RCL - M  
 F004 RCL  $\div$  S  
 F005  $\times 2$   
 F006 2  
 F007  $\div$   
 F008  $+/-$   
 F009  $e^X$

**프로그램 줄:**  
**(RPN 모드)**

**설명**

F010 RTN           호출 루틴으로 돌아갑니다.  
체크섬 및 길이: B3EB 31

**사용되는 플래그:**

없음

**참고:**

이 프로그램의 정확성은 디스플레이 설정에 따라 달라집니다. 입력이  $\pm 3$  표준 편차 사이의 영역에 있는 경우 4개 이상의 유효 숫자 표시가 대부분의 응용 프로그램에 적합합니다.

전체 자릿수에서는 입력 제한이  $\pm 5$  표준 편차가 됩니다. 계산 시간은 표시되는 자릿수가 적을수록 훨씬 단축됩니다.

루틴 Q에서 상수 0.5는 2와  $\sqrt{1/x}$ 로 대체될 수 있습니다.

역 기능을 사용하지 않으려는 경우 루틴 I와 T에 역 루틴을 입력할 필요가 없습니다.

**프로그램 명령:**

1. 프로그램 루틴을 입력하고, 완료되면 **[C]**를 누릅니다.
2. **[XEQ] [S] [ENTER]**를 누릅니다.
3. M을 묻는 프롬프트가 표시되면 모집단 평균을 입력하고 **[R/S]**를 누릅니다. 평균이 0이면 **[R/S]**만 누릅니다.
4. S를 묻는 프롬프트가 표시되면 모집단 표준 편차를 입력하고 **[R/S]**를 누릅니다. 표준 편차가 1이면 **[R/S]**만 누릅니다.
5. Q(X)가 제공된 경우 X를 계산하려면 이러한 명령의 9단계로 건너뜁니다.
6. X가 제공된 경우 Q(X)를 계산하려면 **[XEQ] [D] [ENTER]**를 누릅니다.
7. 프롬프트가 표시되면 X의 값을 입력하고 **[R/S]**를 누릅니다. 결과인 Q(X)가 표시됩니다.
8. 동일한 평균과 표준 편차를 사용하여 새로운 X의 Q(X)를 계산하려면 **[R/S]**를 누르고 7단계로 이동합니다.

9.  $Q(X)$ 가 제공된 경우  $X$ 를 계산하려면 **[XEQ] [1] [ENTER]**를 누릅니다.
10. 프롬프트가 표시되면  $Q(X)$ 의 값을 입력하고 **[R/S]**를 누릅니다. 결과인  $X$ 가 표시됩니다.
11. 동일한 평균과 표준 편차를 사용하여 새로운  $Q(X)$ 의  $X$ 를 계산하려면 **[R/S]**를 누르고 10단계로 이동합니다.

### 사용되는 변수:

$D$	더미 적분 변수입니다.
$M$	모집단 평균으로, 기본값은 0입니다.
$Q$	위쪽 꼬리 영역에 해당하는 확률입니다.
$S$	모집단 표준 편차로, 기본값은 1입니다.
$T$	값 $S \times \sqrt{2\pi}$ 를 역 프로그램으로 전달하기 위해 일시적으로 사용되는 변수입니다.
$X$	위쪽 꼬리 영역의 왼쪽을 정의하는 입력 값입니다.

### 예 1:

친한 친구가 소개팅 상대의 지능이 " $3\sigma$ "라고 알려줍니다. 이 말을 듣고 소개팅 상대가 평균보다 표준 편차가 3 이상 차이가 나는 사람들을 제외한 국내 인구보다 지적이라고 해석합니다.

국내 인구에 10,000명의 가능한 소개팅 상대가 있다고 가정할 경우 이 중에서 몇 명이 " $3\sigma$ " 범위에 포함되겠습니까? 이 문제는 표준 편차의 측면에서 설명되기 때문에  $M$ 에 기본값 0,  $S$ 에 1을 사용합니다.

키:	디스플레이:	설명:
<b>(RPN 모드)</b>		
<b>[XEQ] [S] [ENTER]</b>	$M?$ 0.0000	초기화 루틴을 시작합니다.
<b>[R/S]</b>	$S?$ 1.0000	$M$ 에 대해 기본값 0을 적용합니다.
<b>[R/S]</b>	1.0000	$S$ 에 대해 기본값 1을 적용합니다.
<b>[XEQ] [D] [ENTER]</b>	$X?$ 값	분포 프로그램을 시작하고 $X$ 를 묻는 프롬프트를 표시합니다.

**3** **R/S**

Q=  
0.0013

$X$ 에 3을 입력하고  $Q(X)$  계산을 시작합니다. 평균에서 표준 편차 3 이내의 모든 사람보다 작은 인구 비율을 표시합니다.

**1** **0** **0** **0** **0** **0**

13.4984

**X**

인구 수를 곱합니다. 로컬 인구 중에서 조건에 맞는 소개팅 상대 수를 대략적으로 표시합니다.

친구가 때때로 과장해서 말한다는 것을 알고 있기 때문에 " $2\sigma$ " 소개팅 상대가 얼마나 드문지 확인하려고 합니다. **R/S**만 누르면 프로그램을 다시 실행할 수 있습니다.

**키:**  
**(RPN 모드)**

**디스플레이:**

**설명:**

**R/S**

X?  
3.0000

프로그램을 다시 시작합니다.

**2** **R/S**

Q=  
0.0228

$X$ 값 2를 입력하고  $Q(X)$ 를 계산합니다.

**1** **0** **0** **0** **0** **0**

227.5012

인구 수를 곱하여 수정된 추정값을 구합니다.

**X**

## 예 2:

테스트 점수 집합의 평균은 55입니다. 표준 편차는 15.3입니다. 표준 정규 곡선이 분포를 적절하게 모델링한다고 가정할 경우 무작위로 선택한 학생의 점수가 90점 이상일 확률은 얼마입니까? 10%의 학생만 더 높은 점수를 받는다면 예상되는 점수는 몇 점입니까? 20%의 학생만 더 낮은 점수를 받는다면 예상되는 점수는 몇 점입니까?

키:	디스플레이:	설명:
(RPN 모드)		
XEQ [S] ENTER	M?	초기화 루틴을 시작합니다.
[5] [5] [R/S]	0.0000	
[1] [5] [.] [3] [R/S]	S?	55를 평균으로 저장합니다.
XEQ [D] ENTER	1.0000	
	15.3000	15.3을 표준 편차로 저장합니다.
	X?	분포 프로그램을 시작하고 X를 묻는
	Q?	프롬프트를 표시합니다.
[9] [0] [R/S]	Q=	X에 90을 입력하고 Q(X)를 계산합니다.
	0.0111	

따라서 약 1% 학생만 90점보다 더 높은 점수를 받을 것으로 예상합니다.

키:	디스플레이:	설명:
(RPN 모드)		
XEQ [I] ENTER	Q?	역 루틴을 시작합니다.
[0] [.] [1] [R/S]	0.0111	
	X=	0.1(10%)을 Q(X)에 저장하고 X를 계
	74.6077	산합니다.
[R/S]	Q?	역 루틴을 다시 시작합니다.
	0.1000	
[0] [.] [8] [R/S]	X=	0.8(100%-20%)을 Q(X)에 저장하고 X
	42.1232	를 계산합니다.

---

## 그룹화된 표준 편차

그룹화된 데이터의 표준 편차인  $S_{xy}$ 는 데이터 요소  $x_1, x_2, \dots, x_n$ 의 표준 편차입니다. 데이터 요소는 양의 정수 빈도  $f_1, f_2, \dots, f_n$ 로 발생합니다.

$$S_{xy} = \sqrt{\frac{\sum x_i^2 f_i - \frac{(\sum x_i f_i)^2}{\sum f_i}}{(\sum f_i) - 1}}$$

이 프로그램을 사용하여 데이터를 입력하고, 입력을 수정하고, 그룹화된 데이터의 표준 편차와 가중 평균을 계산할 수 있습니다.



**프로그램 목록:**

<b>프로그램 줄: (ALG 모드)</b>	<b>설명</b>
S001 LBL S	그룹화된 표준 편차 프로그램을 시작합니다.
S002 CLΣ	통계 레지스터(-27에서 -32 사이)를 지웁니다.
S003 0	
S004 STO N	카운트 N을 지웁니다.
체크섬 및 길이: E5BC 13	
I001 LBL I	통계 데이터 요소를 입력합니다.
I002 INPUT X	데이터 요소를 X에 저장합니다.
I003 INPUT F	데이터 요소 빈도를 F에 저장합니다.
I004 1	N의 증분을 입력합니다.
I005 STO B	
I006 RCL F	데이터 요소 빈도 f를 다시 호출합니다.
체크섬 및 길이: 3387 19	
F001 LBL F	합계를 누적합니다.
F002 -27	
F003 STO I	레지스터 -27의 색인을 저장합니다.
F004 RCL F	
F005 STO+(I)	레지스터 -27의 $\sum f_i$ 를 업데이트합니다.
F006 RCL× X	$x_i f_i$
F007 STO Z	
F008 -28	
F009 STO I	레지스터 -28의 색인을 저장합니다.
F010 RCL Z	
F011 STO+(I)	레지스터 -28의 $\sum x_i^2 f_i$ 를 업데이트합니다.
F012 RCL× X	$x_i^2 f_i$
F013 STO Z	레지스터 -30의 색인을 저장합니다.
F014 -30	
F015 STO I	
F016 RCL Z	
F017 STO+(I)	레지스터 -30의 $\sum x_i^2 f_i$ 를 업데이트합니다.
F018 RCL B	
F019 STO+ N	N이 증가 또는 감소합니다.
F020 RCL N	
F021 RCL F	
F022 ABS	
F023 STO F	
F024 VIEW N	현재 데이터 쌍의 개수를 표시합니다.

**프로그램 줄:**  
**(AIG 모드)**

**설명**

F025 GTO I001      다음 데이터 입력을 위해 레이블 줄 번호 /로 이동합니다.  
체크섬 및 길이: F6CB 84

G001 LBL G      그룹화된 데이터의 통계를 계산합니다.  
G002 s<sub>x</sub>      그룹화된 표준 편차입니다.

G003 STO S  
G004 VIEW S      그룹화된 표준 편차를 표시합니다.  
G005  $\bar{x}$       가중 평균입니다.

G006 STO M  
G007 VIEW M      가중 평균을 표시합니다.  
G008 GTO I001      추가 요소를 입력하려면 뒤로 이동합니다.  
체크섬 및 길이: DAF2 24

U001 LBL U      데이터 입력 오류를 실행 취소합니다.  
U002 -1      N의 감량을 입력합니다.

U003 STO B  
U004 RCL F      마지막 데이터 빈도 입력을 다시 호출합니다.  
U005 +/-      f의 부호를 변경합니다.

U006 STO F  
U007 GTO F001      카운트와 합계를 조정합니다.  
체크섬 및 길이: 03F4 23

## 사용되는 플래그:

없음

## 프로그램 명령:

1. 프로그램 루틴을 입력하고, 완료되면 **[C]**를 누릅니다.
2. **[XEQ] [S] [ENTER]**를 눌러 새 데이터 입력을 시작합니다.
3.  $x_i$  값(데이터 요소)을 입력하고 **[R/S]**를 누릅니다.
4.  $f_i$  값(빈도)을 입력하고 **[R/S]**를 누릅니다.
5. 입력한 요소 수를 확인한 후 **[R/S]**를 누릅니다.
6. 각 데이터 요소에 대해 3-5단계를 반복합니다.

4단계에서 **[R/S]**를 누른 후 데이터 입력 오류( $x_i$  또는  $f_i$ )를 발견하게 되면 **[XEQ] [U] [ENTER]**를 누른 다음 **[R/S]**를 다시 누릅니다. 그런 후에 3단계로 돌아가서 올바른 데이터를 입력합니다.

7. 마지막 데이터 쌍이 입력되면 **[XEQ] [G] [ENTER]**를 눌러 그룹화된 표준 편차를 계산하고 표시합니다.
8. **[R/S]**를 눌러 그룹화된 데이터의 가중 평균을 표시합니다.
9. 데이터 요소를 더 추가하려면 **[R/S]**를 누르고 3단계에서 계속합니다.  
새 문제를 시작하려면 2단계에서 시작합니다. 사용되는 변수:

$X$	데이터 요소입니다.
$F$	데이터 요소 빈도입니다.
$N$	데이터 쌍 카운터입니다.
$S$	그룹화된 표준 편차입니다.
$M$	가중 평균입니다.
$i$	올바른 통계 레지스터의 주소를 간접적으로 지정하는 데 사용되는 색인 변수입니다.
레지스터 -27	합계 $\Sigma f$ 입니다.
레지스터 -28	합계 $\Sigma x_i f_i$ 입니다.
레지스터 -30	합계 $\Sigma x_i^2 f_i$ 입니다.

예:

다음 데이터를 입력하고 그룹화된 표준 편차를 계산합니다.

그룹	1	2	3	4	5	6
$x_i$	5	8	13	15	22	37
$f_i$	17	26	37	43	73	115

키:	디스플레이:	설명:
(ALG 모드)		
<b>XEQ</b> <b>S</b> <b>ENTER</b>	X?	첫 번째 $x_i$ 를 묻는 프롬프트를 표시합니다.
<b>5</b> <b>R/S</b>	5.0000	5를 X에 저장하고 첫 번째 $f_i$ 를 묻는 프롬프트를 표시합니다.
<b>17</b> <b>R/S</b>	17.0000	17을 F에 저장하고 카운터를 표시합니다.
<b>R/S</b>	X?	두 번째 $x_i$ 를 묻는 프롬프트를 표시합니다.
<b>8</b> <b>R/S</b>	8.0000	두 번째 $f_i$ 를 묻는 프롬프트를 표시합니다.
<b>26</b> <b>R/S</b>	26.0000	카운터를 표시합니다.
<b>R/S</b>	X?	세 번째 $x_i$ 를 묻는 프롬프트를 표시합니다.
<b>13</b> <b>R/S</b>	13.0000	세 번째 $f_i$ 를 묻는 프롬프트를 표시합니다.
<b>43</b> <b>R/S</b>	43.0000	카운터를 표시합니다.

$x_3$ 에 13대신 14를 입력하여 오류를 만들었습니다. 루틴 U를 실행하여 오류를 실행 취소합니다.

**XEQ U ENTER**

N=  
2.0000

잘못된 데이터를 제거하고 수정된 카운터를 표시합니다.

**R/S**

X?  
14.0000

새로운 세 번째  $x_i$ 를 묻는 프롬프트를 표시합니다.

**1 3 R/S**

F?  
37.0000

새로운 세 번째  $f_i$ 를 묻는 프롬프트를 표시합니다.

**R/S**

N=  
3.0000

카운터를 표시합니다.

**R/S**

X?  
13.0000

네 번째  $x_i$ 를 묻는 프롬프트를 표시합니다.

**1 5 R/S**

F?  
37.0000

네 번째  $f_i$ 를 묻는 프롬프트를 표시합니다.

**XEQ U ENTER**

N=  
4.0000

카운터를 표시합니다.

**R/S**

X?  
15.0000

다섯 번째  $x_i$ 를 묻는 프롬프트를 표시합니다.

**1 3 R/S**

F?  
43.0000

다섯 번째  $f_i$ 를 묻는 프롬프트를 표시합니다.

**R/S**

N=  
5.0000

카운터를 표시합니다.

**R/S**

X?  
22.0000

여섯 번째  $x_i$ 를 묻는 프롬프트를 표시합니다.

**1 5 R/S**

F?  
73.0000

여섯 번째  $f_i$ 를 묻는 프롬프트를 표시합니다.

**XEQ U ENTER**

N=  
6.0000

카운터를 표시합니다.

**R/S**

S=  
11.4118

여섯 번째 데이터 요소의 그룹화된 표준 편차( $s_x$ )를 계산하고 표시합니다.

**1 3 R/S**

M=  
23.4084

가중 평균( $\bar{X}$ )을 계산하고 표시합니다.

**R/S**

23.4084

VIEW를 지원합니다.



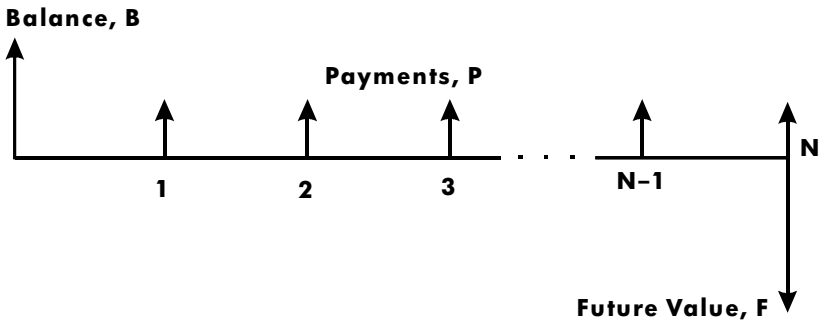
## 기타 프로그램 및 방정식

### 화폐의 시간적 가치

TVM("화폐의 시간적 가치 방정식")의 5개 값 중 4개가 제공되면 5번째 값을 구할 수 있습니다. 이 방정식은 소비자 및 가계 대출과 저축 계좌 등의 다양한 금융 응용 프로그램에서 유용합니다.

TVM 방정식은 다음과 같습니다.

$$P \left[ \frac{1 - (1 + I/100)^{-N}}{I/100} \right] + F(1 + (I/100))^N + B = 0$$



현금 금액(잔액  $B$ , 상환액  $P$ , 미래 잔액  $F$ )의 부호는 현금 흐름의 방향과 일치합니다. 받은 금액에는 양의 부호가 지정되고 상환한 금액에는 음의 부호가 지정됩니다. 모든 문제를 두 가지 측면에서 볼 수 있습니다. 대출자와 차용자는 동일한 문제를 반대 부호로 봅니다.

## 방정식 입력:

다음 방정식을 입력합니다.

$$P \times 100 \times (1 - (1 + I \div 100)^{-N}) \div I + F \times (1 + I \div 100)^{-N} + B$$

키: (RPN 모드)	디스플레이:	설명:
<b>[EQN]</b>	EQN LIST TOP 또는 현재 방정식 P×100_	방정식 모드를 선택합니다. 방정식 입력을 시작합니다.
<b>[RCL] [P] [×] [1] [0] [0]</b>	P×100×(1_)	
<b>[×] [( )] [1] [-]</b>	P×100×(1-(+_))	
<b>[( )] [1] [+]</b>	P×100×(1-(1+_))	
<b>[RCL] [I] [÷] [1] [0] [0]</b>	←0×(1-(1+I÷100)_ )→	
<b>[&gt;] [y<sup>x</sup>]</b>	←(1-(1+I÷100)^_ )→	
<b>[+/-] [RCL] [N] [&gt;]</b>	←(1+I÷100)^-N)_	
<b>[÷] [RCL] [I] [+]</b> <b>[RCL] [F]</b>	←100)^-N)÷I+F×_	
<b>[×]</b>		
<b>[( )] [1] [+]</b> <b>[RCL] [I]</b>	←^-N)÷I+F×(1+I)_	
<b>[÷] [1] [0] [0] [&gt;]</b>	←I+F×(1+I÷100)_	
<b>[y<sup>x</sup>] [+/-] [RCL] [N]</b>	←×(1+I÷100)^-N)_	
<b>[+] [RCL] [B]</b>	←1+I÷100)^-N+B_	
<b>[ENTER]</b>	P×100×(1-(1+I÷	➡ 방정식을 종료합니다.
<b>[↵] [SHOW] (누르고 있음)</b>	CK=CEFA LN=41	체크섬 및 길이입니다.

## 참고:

TVM 방정식에서 **DIVIDE BY 0** 오류를 방지하려면 /가 0이 아니어야 합니다.

의 값을 구하고 있으며 현재 값을 모르는 경우 SOLVE 계산(**[1] [↵] [STO] [1]**)을 시작하기 전에 **[↵] [SOLVE] [1]**를 누릅니다.



값을 묻는 프롬프트가 표시되는 순서는 값을 구하는 변수에 따라 달라집니다.

### SOLVE 명령:

1. 첫 번째 TVM 계산이 이율 I의 값을 구하는 것이면 **1** **↵** **STO** **1**를 누릅니다.
2. **EQN**을 누릅니다. 필요한 경우 **▲** 또는 **▼**를 눌러 TVM 방정식에 도달할 때까지 방정식을 스크롤합니다.
3. 다음 5개 작업 중 하나를 수행합니다.

- a. **↵** **SOLVE** **N**을 눌러 복리 기간 수를 계산합니다.
- b. **↵** **SOLVE** **I**를 눌러 정기 이자를 계산합니다.

매월 상환하는 경우 **예** 대해 반환된 결과는 **월**이율 **이**입니다. 연이율을 보려면 **12** **X**를 누릅니다.

- c. **↵** **SOLVE** **B**를 눌러 대출 또는 저축 계좌의 초기 잔액을 계산합니다.
  - d. **↵** **SOLVE** **P**를 눌러 정기 상환액을 계산합니다.
  - e. **↵** **SOLVE** **F**를 눌러 **대출**의 미래 금액 또는 잔액을 계산합니다.
4. 해당 프롬프트가 표시되면 알려진 4개 변수의 값을 입력하고, 각 값을 입력한 후 **R/S**를 누릅니다.
  5. 마지막 **R/S**를 누르면 미지수 변수의 값이 계산되어 표시됩니다.
  6. 새로운 변수를 계산하거나 동일한 변수를 다른 데이터로 다시 계산하려면 2단계로 돌아갑니다.

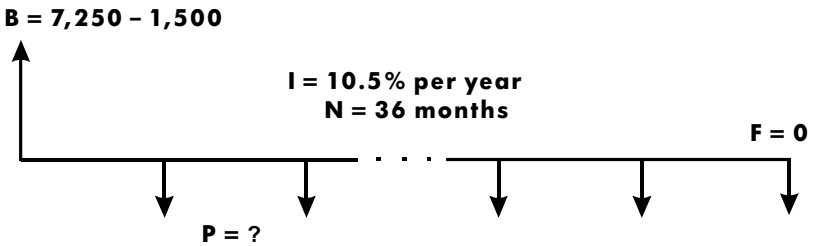
SOLVE는 초기 추정값 없이 이 응용 프로그램에서 효과적으로 작동합니다.

**사용되는 변수:**

- $N$       복리 기간 수입입니다.
- $I$       백분율로 표시되는 정기 이율입니다. 예를 들어, 연이율이 15%이고 매년 12회 상환하는 경우 정기 이율  $i$ 는  $15 \div 12 = 1.25\%$ 입니다.)
- $B$       대출 또는 저축 계좌의 초기 잔액입니다.
- $P$       정기 상환액입니다.
- $F$       저축 계좌의 미래 금액 또는 대출 잔액입니다.

**예:**

**1부.** 월 복리로 계산되는 10.5%의 연이율로 3년(36개월)간 대출하여 자동차를 구입하려고 합니다. 자동차의 구입 가격은 7,250달러입니다. 정기 상환액은 1,500달러입니다.



**키:**  
(RPN 모드)

**디스플레이:**

**설명:**

**DISPLAY** **1** (FIX) **2**

FIX 2 디스플레이 형식을 선택합니다.

**EQN** (필요에 따라  $\sqrt{x}$ )  $P \times 100 \times (1 - (1 + I)^{-N})$

TVM 방정식의 맨 왼쪽 부분을 표시합니다.

**→ SOLVE** **P**

I?  
↵

P를 선택하고 /를 묻는 프롬프트를 표시합니다.

**1 0 . 5** **ENTER**

I?

연이율 입력 값을 이에 해당하는 월이율로 변환합니다.

**1 2** **÷**

0.88

**R/S**

N?  
↵

0.88을  $i$ 에 저장하고 N을 묻는 프롬프트를 표시합니다.

**3 6** **R/S**

F?  
↵

36을 N에 저장하고 F를 묻는 프롬프트를 표시합니다.

**0** **R/S**

B?

0을 *F*에 저장하고 *B*를 묻는 프롬프트를 표시합니다.

**7** **2** **5** **0** **ENTER**

값

시작 대출 잔액인 *B*를 계산합니다.

**1** **5** **0** **0** **-**

B?

5,750.00

**R/S**

SOLVING

5750을 *B*에 저장하고 월 상환액 *P*를 계산합니다.

P=

-186.89

차용자의 관점에서 대출을 보았기 때문에 결과가 음수입니다. 차용자가 받은 금액(시작 잔액)은 양수이고 상환된 금액은 음수입니다.

## 2부. 월 상환액을 10달러 줄이는 이율은 얼마입니까?

**키:**  
**(RPN 모드)**

**디스플레이:**

**설명:**

**[EQN]**

$P \times 100 \times (1 - (1 + I)^{-N}) \Rightarrow$  TVM 방정식의 맨 왼쪽 부분을 표시합니다.

**[>] SOLVE [I]**

P?

*I*를 선택하고 *P*를 묻는 프롬프트를 표시합니다.

-186.89

**[>] [RND]**

P?

상환액을 소수 두 자릿수까지 반올림합니다.

-186.89

**[1] [0] [+]**

P?

새 상환액을 계산합니다.

-176.89

**[R/S]**

N?

-176.89를 *F*에 저장하고 *N*을 묻는 프롬프트를 표시합니다.

36.00

**[R/S]**

F?

36을 *N*에 유지하고 *F*를 묻는 프롬프트를 표시합니다.

0.00

**[R/S]**

B?

0을 *F*에 유지하고 *B*를 묻는 프롬프트를 표시합니다.

5,750.00

**[R/S]**

SOLVING

5750을 *B*에 유지하고 월이율을 계산합니다.

I=

0.56

**[1] [2] [X]**

6.75

연이율을 계산합니다.

**3부.** 계산된 이율(6.75%)을 사용하여 2년 후에 자동차를 매도한다고 가정합니다. 상환해야 하는 잔액은 얼마입니까? 즉, 2년 후의 미래 잔액은 얼마입니까?

2부의 이율  $r$ 은 0이 아니므로 새로운  $r$ 을 계산할 때 **DIVIDE BY 0** 오류가 발생하지 않습니다.

키: (RPN 모드)	디스플레이:	설명:
<b>[EQN]</b>	$P \times 100 \times (1 - (1 + I)^{-N}) \div I$	TVM 방정식의 맨 왼쪽 부분을 표시합니다.
<b>[<math>\rightarrow</math>] [SOLVE] [F]</b>	P? -176.89	$r$ 을 선택하고 $r$ 을 묻는 프롬프트를 표시합니다.
<b>[R/S]</b>	I? 0.56	$r$ 을 유지하고 $r$ 을 묻는 프롬프트를 표시합니다.
<b>[R/S]</b>	N? 36.00	0.56을 $r$ 에 유지하고 $N$ 을 묻는 프롬프트를 표시합니다.
<b>[2] [4] [R/S]</b>	B? 5,750.00	24를 $N$ 에 저장하고 $B$ 를 묻는 프롬프트를 표시합니다.
<b>[R/S]</b>	SOLVING F= -2,047.05	5750을 $B$ 에 유지하고 미래 잔액인 $F$ 를 계산합니다. 부호가 음수이므로 이 금액을 상환해야 함을 나타냅니다.
<b>[<math>\leftarrow</math>] [DISPLAY] [1]</b> (FIX) <b>[4]</b>		FIX 4 디스플레이 형식을 설정합니다.

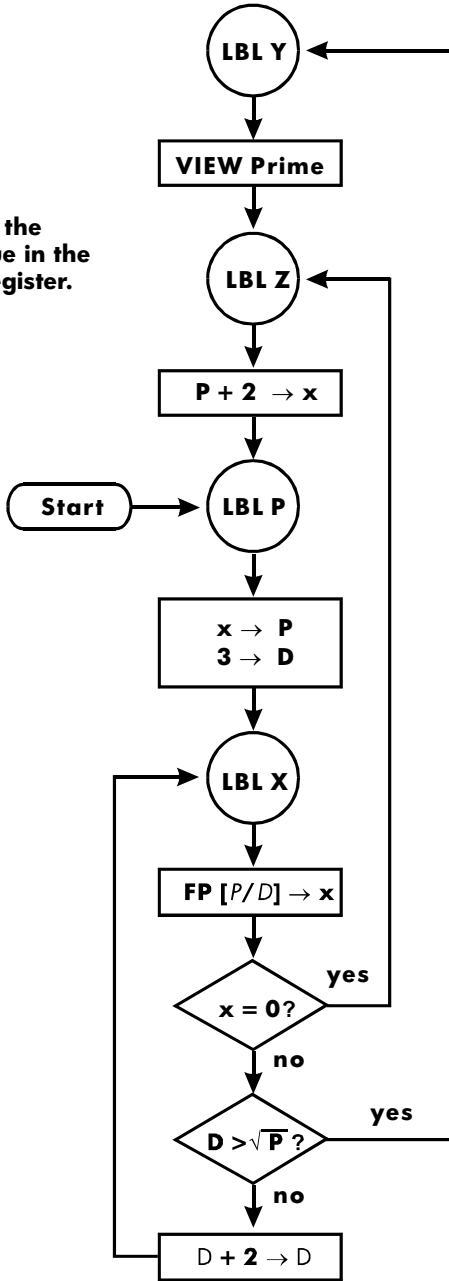
---

## 소수 생성기

이 프로그램에서는 3보다 큰 양의 정수를 받아들입니다. 숫자가 소수(자신과 1 이외의 정수로 나눌 수 없는 수)이면 프로그램에서 입력 값을 반환합니다. 입력 값이 소수가 아닌 경우 입력 값보다 큰 첫 번째 소수가 반환됩니다.

프로그램은 가능한 모든 인수를 시도하여 소수가 아닌 숫자를 식별합니다. 숫자가 소수가 아닌 경우 프로그램에서 2를 더하고 값이 여전히 홀수인지 확인한 다음 테스트하여 소수를 찾았는지 확인합니다. 이 프로세스는 소수를 찾을 때까지 계속됩니다.

Note:  $x$  is the value in the X-register.



## 프로그램 목록:

프로그램 줄: (ALG 모드)	설명
Y001 LBL Y	이 루틴에서는 소수 $P$ 를 표시합니다.
Y002 VIEW P	
체크섬 및 길이: 2CC5 6	
Z001 LBL Z	이 루틴에서는 $P$ 에 2를 더합니다.
Z002 2+ P	
체크섬 및 길이: EFB2 9	
P001 LBL P	이 루틴에서는 $P$ 의 입력 값을 저장합니다.
P002 LASTx▸ P	
P003 FP(P÷2)	
P004 x<>v	
P005 0	
P006 x=v?	입력 값이 짝수인지 테스트합니다.
P007 1+P▸P	입력 값이 짝수이면 $P$ 를 증가시킵니다.
P008 3▸D	3을 테스트 제수 $D$ 에 저장합니다.
체크섬 및 길이: EA89 47	
X001 LBL X	이 루틴에서는 $P$ 를 테스트하여 소수인지 확인합니다.
X002 FP(P÷D)	
X003 x=0?	나머지가 0인지 테스트합니다(소수가 <i>아님</i> ).
X004 GTO Z001	숫자가 소수가 아니면 다음 가능성을 시도합니다.
X005 SQRT(P)	
X006 x<>v	
X007 D	
X008 x>v?	테스트하여 가능한 모든 인수가 시도되었는지 확인합니다.
X009 GTO Y001	모든 인수가 시도된 경우 디스플레이 루틴으로 분기합니다.
X010 2+D▸D	
X011 GTO X001	분기하여 새 인수로 잠재적 소수를 테스트합니다.
체크섬 및 길이: C6B5 53	

### 사용되는 플래그:

없음

### 프로그램 명령:

1. 프로그램 루틴을 입력하고, 완료되면 **C**를 누릅니다.
2. 3보다 큰 양의 정수를 입력합니다.
3. **XEQ** **P** **ENTER**를 눌러 프로그램을 실행합니다. 소수  $P$ 가 표시됩니다.
4. 다음 소수를 확인하려면 **R/S**를 누릅니다.

### 사용되는 변수:

$P$  소수 및 잠재적 소수입니다.  
 $D$   $P$ 의 현재 값을 테스트하는 데 사용되는 제수입니다.

### 참고:

입력 값이 3보다 큰지 확인하는 테스트는 수행되지 않습니다.

### 예:

789 뒤의 첫 번째 소수는 무엇입니까? 다음 소수는 무엇입니까?

키:	디스플레이:	설명:
(ALG 모드)		
<b>7</b> <b>8</b> <b>9</b> <b>XEQ</b>	P=	789 뒤의 다음 소수를 계산합니다.
<b>P</b> <b>ENTER</b>	797.0000	
<b>R/S</b>	P=	797 뒤의 다음 소수를 계산합니다.
	809.0000	



---

## 벡터의 외적

다음은 프로그램 함수를 사용하여 외적을 계산하는 방법을 보여 주는 예입니다.

외적:

$$\mathbf{v}_1 \times \mathbf{v}_2 = (YW - ZV)\mathbf{i} + (ZU - XW)\mathbf{j} + (XV - YU)\mathbf{k}$$

여기서

$$\mathbf{v}_1 = Xi + Yj + Zk$$

및

$$\mathbf{v}_2 = Ui + Vj + Wk$$

### 프로그램 줄: (RPN 모드)

```
R001 LBL R
R002 INPUT X
R003 INPUT Y
R004 INPUT Z
R005 GTO R001
```

체크섬 및 길이: D82E 15

```
E001 LBL E
E002 RCL X
E003 STO U
E004 RCL Y
E005 STO V
E006 RCL Z
E007 STO W
E008 GTO R001
```

체크섬 및 길이: B6AF 24

### 설명

직사각형 입력/디스플레이 루틴의 시작을 정의합니다.

X의 입력 값을 표시하거나 받아들입니다.

Y의 입력 값을 표시하거나 받아들입니다.

Z의 입력 값을 표시하거나 받아들입니다.

R001로 이동하여 벡터를 입력합니다.

벡터 입력 루틴의 시작을 정의합니다.

X, Y 및 Z의 값을 각각 U, V 및 W에 복사합니다.

R001로 이동하여 벡터를 입력합니다.

**프로그램 줄:**  
**(RPN 모드)**

**설명**

C001 LBL C	외적 루틴의 시작을 정의합니다.
C002 RCL Y	
C003 RCL×W	
C004 RCL Z	
C005 RCL×V	
C006 -	X성분인 $(YW - ZV)$ 를 계산합니다.
C007 STO A	
C008 RCL Z	
C009 RCL×U	
C010 RCL X	
C011 RCL×W	
C012 -	Y성분인 $(ZU - WX)$ 를 계산합니다.
C013 STO B	
C014 RCL X	
C015 RCL×V	
C016 RCL Y	
C017 RCL×U	
C018 -	
C019 STO Z	Z성분인 $(XV - YU)$ 를 저장합니다.
C020 RCL A	
C021 STO X	X성분을 저장합니다.
C022 RCL B	
C023 STO Y	Y성분을 저장합니다.
C024 GTO R001	R001로 이동하여 벡터를 입력합니다.

체크섬 및 길이: 838D 72

**예:**

두 벡터  $v_1=2i+5j+4k$ 와  $v_2=i-2j+3k$ 의 외적을 계산합니다.

키:	디스플레이:	설명:
<b>XEQ</b> <b>R</b> <b>ENTER</b>	X?	R 루틴을 실행하여 벡터 값을 입력합니다.
<b>1</b> <b>R/S</b>	v?	x 성분의 v2를 입력합니다.
<b>2</b> <b>+/-</b> <b>R/S</b>	z?	y 성분의 v2를 입력합니다.
<b>3</b> <b>R/S</b>	X?	z 성분의 v2를 입력합니다.
<b>XEQ</b> <b>E</b> <b>ENTER</b>	X?	1 E 루틴을 실행하여 U, V 및 W
<b>2</b> <b>R/S</b>	v?	1 변수의 v2를 교환합니다.
<b>5</b> <b>R/S</b>	z?	x 성분의 v1을 입력합니다.
<b>4</b> <b>R/S</b>	X?	-2 y 성분의 v1을 입력합니다.
<b>XEQ</b> <b>C</b> <b>ENTER</b>	X?	3 z 성분의 v1을 입력합니다.
<b>R/S</b>	v?	2 C 루틴을 실행하여 외적의 x 성분
<b>R/S</b>	z?	23 을 계산합니다.
		외적의 y 성분을 계산합니다.
		-2 외적의 z 성분을 계산합니다.
		-9



# 3부

## 부록 및 참조

---



# 지원, 배터리 및 서비스

---

## 계산기 지원

계산기 지원 부서에 문의하시면 계산기 사용 질문에 대한 답변을 받을 수 있습니다. 다음 절에 나와 있는 "일반적인 질문에 대한 답변"에서 많은 고객분들이 문의하신 유사한 내용의 질문을 확인할 수 있습니다. 여기에 여러분의 질문에 대한 답변이 없으면 A-8페이지에 나와 있는 계산기 지원 부서에 문의하십시오.

### 일반적인 질문에 대한 답변



Q: 계산기가 제대로 작동하고 있는지 어떻게 판단할 수 있습니까?

A: 자가 진단 테스트에 대한 설명이 들어 있는 A-5페이지를 참조하십시오.



Q: 숫자의 소수점이 마침표가 아닌 쉼표로 표시되어 있습니다. 마침표를 복원하려면 어떻게 해야 합니까?

A:    (5.) 함수를 사용합니다(1-23페이지).

Q: 디스플레이에 표시되는 소수 자릿수를 변경하려면 어떻게 해야 합니까?

A:   메뉴를 사용합니다(1-21페이지).

Q: 메모리 전체 또는 일부를 지우려면 어떻게 해야 합니까?

A:  를 선택하면 CLEAR 메뉴가 표시됩니다. 이 메뉴를 사용하면 x(X 레지스터의 숫자), 모든 직접 변수, 모든 메모리, 모든 통계 데이터, 모든 스택 수준 및 모든 간접 변수를 지울 수 있습니다.

Q: 숫자에서 "E"는 무엇을 의미합니까(예:  $2.51E-13$ )?

A: 10의 지수(예:  $2.51 \times 10^{-13}$ )입니다.


Q: 계산기에 MEMORY FULL이라는 메시지가 표시되는 경우 어떻게 해야 하나요?

A: 계속하기 전에 메모리 일부를 지워야 합니다. 부록 B를 참조하십시오.

Q:  $\pi$  라디안의 사인 또는 탄젠트를 계산할 때 0이 아닌 아주 작은 숫자가 표시되는 이유는 무엇입니까?

A:  $\pi$ 는 정확하게 계산기의 12자리 전체 자릿수로 나타낼 수 없습니다.

Q: 삼각 함수를 사용할 때 잘못된 답이 나오는 이유는 무엇입니까?

A: 계산기에서 올바른 각도 모드( 1DEG, 2RAD 또는 3GRD)를 사용하고 있는지 확인해야 합니다.

Q: 디스플레이에 있는 표시기는 무엇을 의미합니까?

A: 표시기는 계산기 상태에 대한 정보를 나타냅니다. 1장의 "표시기"를 참조하십시오.

Q: 숫자가 분수로 표시됩니다. 10진수를 얻으려면 어떻게 해야 하나요?

A:  (FDISP)를 누릅니다.

---

## 환경적인 제한 사항


제품 안정성을 유지하려면 다음과 같은 온도 및 습도 제한을 유지해야 합니다.

- 작동 온도: 0 ~ 45°C(32 ~ 113°F)
- 보관 온도: -20 ~ 65°C(-4 ~ 149°F)
- 작동 및 보관 습도: 최대 40°C(104°F)에서 90% 상대 습도




## 배터리 교체

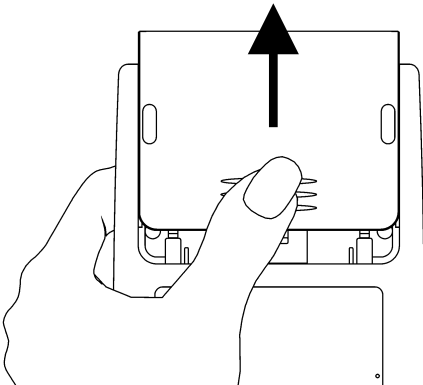
계산기는 동전형 3볼트 리튬 배터리인 CR2032에 의해 전력을 공급받습니다.

배터리 부족 표시기(  )가 나타나면 바로 배터리를 교체하십시오. 배터리 표시기가 켜져 있고 디스플레이가 흐리면 데이터가 손실될 수 있습니다. 데이터가 손실되면 MEMORY CLEAR 메시지가 표시됩니다.

*배터리를 꺼내면 2분 내에 다른 배터리로 교체해야 저장된 정보가 손실되지 않습니다. (따라서 배터리함을 열기 전에 새 배터리를 바로 사용할 수 있도록 준비해 두어야 합니다.)*

### 배터리를 교체하려면

1. 두 개의 새 버튼형 배터리를 준비합니다. 배터리 단자는 건드리지 말고 배터리 가장자리만 잡으십시오.
2. 계산기가 꺼져 있는지 확인합니다. **배터리 교체 절차가 모두 완료되기 전까지는 ON(  )을 누르지 마십시오. 배터리를 꺼냈을 때 계산기가 켜져 있으면 연속 메모리의 내용이 지워집니다.**  
**用BK+B**
3. 계산기를 뒤집어 배터리 덮개를 밀어 엽니다.



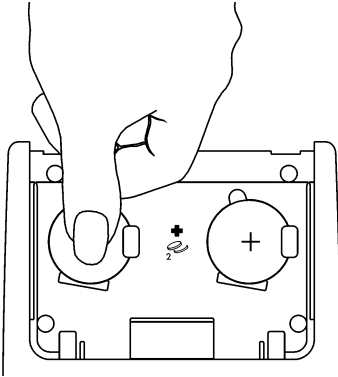
4. 메모리 손실 방지를 위해 사용한 배터리 두 개를 동시에 꺼내지 마십시오. 배터리는 한 번에 한 개씩 꺼내어 교체합니다.

## 경고



배터리를 절단하거나 구멍을 내거나 불 속에 버리지 마십시오. 연소되거나 폭발하여 위험한 화학 물질이 나올 수 있습니다.

5. 양 부호(+)가 바깥을 향하게 새 CR2032 리튬 배터리를 넣습니다.



6. 4-5단계를 참조하여 남은 배터리를 꺼내 교체합니다. 각 배터리의 양 부호(+)가 바깥을 향해야 합니다.
7. 배터리함 덮개를 닫습니다.
8. **[C]**를 누릅니다.

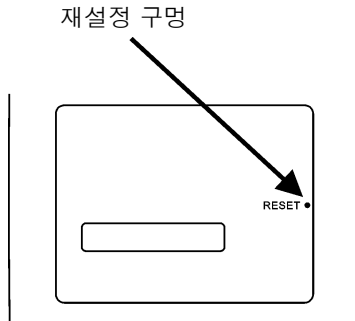
## 계산기 작동 테스트

다음 지침에 따라 계산기가 제대로 작동하는지 확인합니다. 각 단계 이후의 계산기를 테스트하여 해당 작동이 제대로 복원되었는지 확인합니다. 계산기를 수리해야 하는 경우 A-8페이지를 참조하십시오.

### ■ 계산기가 켜지지 않거나(1-4단계) 키를 누를 때 응답하지 않는 경우 (1-3 단계):

1. 계산기를 재설정합니다. **[C]** 키를 누른 채 **[GTO]**를 누릅니다. 이러한 재설정을 위한 키 입력을 여러 번 반복해야 할 수도 있습니다.
2. 메모리를 지웁니다. **[C]**를 누르고 있다가 **[R/S]**와 **[6]**를 모두 누르고 있습니다. 이 세 키를 동시에 놓으면 메모리가 지워지고 **MEMORY CLEAR** 메시지가 표시됩니다.

3. 배터리를 꺼내고("배터리 교체" 참조) 계산기의 두 배터리 접지부분을 동전으로 가볍게 누릅니다. 배터리를 교체한 다음 계산기를 켭니다.  
MEMORY CLEAR가 표시되어야 합니다.
4. 계산기가 키 입력에 여전히 반응하지 않으면 가늘고 뾰족한 물체를 사용하여 RESET 구멍을 누릅니다. 저장되어 있는 데이터는 일반적으로 그대로 유지됩니다.



이렇게 진행했는데 계산기가 제대로 작동하지 않는 경우 수리해야 합니다.

■ **계산기가 키 입력에는 반응하지만 오작동이 의심되는 경우:**

1. 다음 절에서 설명하는 자가 테스트를 진행합니다. 계산기에서 자가 테스트를 통과하지 못한 경우 수리해야 합니다.
2. 계산기에서 자가 테스트를 통과한 경우에는 계산기를 잘못 작동했을 수 있습니다. 설명서의 해당 부분을 다시 읽고 "일반적인 질문에 대한 답변"(A-1페이지)을 확인하십시오.
3. A-8페이지에 나와 있는 계산기 지원 부서에 문의합니다.

---

## 자가 테스트

디스플레이를 켤 수는 있는데 계산기가 제대로 작동하지 않는 것 같으면 다음과 같은 자체 진단 테스트를 수행해 보십시오.

1. **[C]** 키를 누른 채 동시에 **[XEQ]**를 누릅니다.
2. 아무 키나 8번 누르고 나타나는 여러 현상을 살펴 봅니다. 키를 8번 누른 후에는 계산기에서 저작권 메시지 • 2007 HP DEV CO. L. P. 를 표시한 다음 메시지 **KBD 01**을 표시합니다.
3. 다음 순서대로 키를 누릅니다.

**R/S** → **GTO** → **XEQ** → **9** → **^** → **<** → **>** → **RCL** → **R↓** →  
**x↔y** → **6** → **∇** → **SIN** → **COS** → **TAN** → **√x** → **y<sup>x</sup>** → **1/x** →  
**ENTER** → **+/-** → **E** → **4** → **←ENG** → **EQN** → **7** → **8** → **9**  
 → **÷** → **↶** → **4** → **5** → **6** → **x** → **↷** → **1** → **2** → **3** →  
**-** → **C** → **0** → **.** → **Σ+** → **+**

- 올바른 순서대로 키를 누르고 해당 키가 제대로 작동하면 계산기에서 **KBD**와 2자리 숫자를 표시합니다. 계산기에서는 16진수를 사용하여 키를 계산합니다.
- 키를 순서대로 누르지 않거나 키가 제대로 작동하지 않으면 다음 키 입력 시 실패 메시지가 표시됩니다(4단계 참조).

4. 자가 테스트의 결과는 다음 둘 중 하나입니다.

- 계산기가 자가 테스트를 통과하면 **35S-OK**를 표시합니다. 5단계로 이동하십시오.
- 계산기에서 자가 테스트를 통과하지 못하면 **35S-FAIL**과 1자리수를 표시합니다. 키를 순서대로 누르지 않아 이러한 메시지가 표시된 경우에는 계산기를 재설정하고(**C**를 누른 채 **GTO** 누름) 자가 테스트를 다시 실행합니다. 키를 순서대로 눌렀는데 이러한 메시지가 표시된 경우에는 자가 테스트를 반복 실행하여 결과를 확인합니다. 계산기가 다시 자가 테스트를 통과하지 못한 경우에는 수리해야 합니다(A-8페이지 참조). 수리를 위해 계산기를 보낼 때 실패 메시지의 사본도 동봉하십시오.

5. 자가 테스트를 끝내려면 **C**를 누른 채 **GTO**를 눌러 계산기를 재설정합니다.

**C** 및 **9**를 누르면 계산기 출하 시 사용되는 연속 자가 테스트가 시작됩니다. 아무 키나 누르면 이러한 출하 테스트를 중지할 수 있습니다.

---

## 보증

HP 35s 공학용 계산기, 보증 기간: 12개월

1. HP는 제품 구입일로부터 상기 보증 기간 동안 HP 하드웨어, 부속품 및 소모품을 사용할 경우 재질이나 성능에 결함이 없음을 보증합니다. 보증 기간 내 결함이 있는 제품에 대한 통지를 받을 경우 HP는 자체 판단에 따라 결함이 입증된 제품을 수리 또는 교체합니다. 교체할 경우 새 제품 또는 새 제품에 준하는 제품으로 합니다.
2. HP 소프트웨어를 올바르게 설치하여 사용할 경우 제품 구입일로부터 상기 보증 기간 동안 제품의 재질이나 성능의 결함으로 인하여 프로그래밍 명령을 실행하지 못하는 경우는 없음을 보증합니다. 보증 기간 내 결함이 있는 제품에 대한 통지를 받을 경우 HP는 해당 결함으로 인해 프로그래밍 명령을 실행할 수 없는 소프트웨어 미디어를 교체합니다.
3. HP는 HP 제품의 작동에 있어 결함이나 오류가 없음을 보증하지 않습니다. HP가 합당한 시간 내에 보증 조건에 따라 결함 제품을 수리하거나 교체할 수 없는 경우 사용자는 구입 영수증과 함께 제품을 즉시 반납하고 구입 금액을 환불받을 수 있습니다.
4. HP 제품에는 성능면에서 신제품과 동등한 기능의 재생 부품이 포함되어 있을 수 있으며 부수적 용도로 쓰일 수 있습니다.
5. (a) 비정상적 또는 부적절한 유지 보수 또는 조정, (b) HP가 제공하지 않은 소프트웨어, 인터페이스 부속품 및 소모품, (c) 무단 수정 또는 오용, (d) 게시된 환경 사양 외에서의 제품의 작동 또는 (e) 부적절한 현장 환경 또는 유지 보수로 생기는 결함에 대해서는 보증은 적용되지 않습니다.
6. HP는 서면 또는 구두로 다른 명시적 보증 또는 조건을 제공하지 않습니다. 지역 법률에서 허용하는 범위 내에서 상용화, 품질 만족 또는 특정 목적에 대한 적합성과 관련한 묵시적 보증이나 조건은 상기에 명시된 보증 기간으로 제한됩니다. 일부 국가, 구/군, 시/도에서는 묵시적 보증 기간을 제한하는 것을 허용하지 않으므로 위 제한 또는 배제 조항이 사용자에게 적용되지 않을 수도 있습니다. 본 보증은 특정 법적 권리를 제공하며 국가, 구/군, 시/도마다 다른 권리를 제공할 수도 있습니다.

7. 지역 법률이 허용하는 범위 내에서 이 보증서에 명시된 배상 조항은 사용자에게 고유하고 배타적인 권한을 제공합니다. 상기에 명시된 경우를 제외하고 HP 또는 공급업체는 계약, 불법 행위 등에 의해서든 데이터 손실 또는 직접적, 부수적, 파생적(이익 또는 데이터 손실 포함), 특별, 기타 손해에 대하여 어떠한 경우에도 책임을 지지 않습니다. 일부 국가, 구/군, 시/도에서는 부수적 또는 파생적 결함의 배제 또는 제한을 허용하지 않으므로 위 제한 또는 배제 조항이 사용자에게 적용되지 않을 수도 있습니다.
8. HP 제품 및 서비스에 대한 유일한 보증은 제품 및 서비스와 함께 제공되는 명시적 보증서에 규정되어 있습니다. HP는 여기에 포함된 기술적 또는 편집상 오류 또는 누락에 대해 책임지지 않습니다.

**호주 및 뉴질랜드에서 발생한 소비자 거래의 경우 법적 허용 범위 이외에 이 보증서에 들어 있는 보증 조건은 이 제품 판매와 관련된 강행 법규상의 권리를 배제하거나 제한하거나 수정하지 않습니다.**

## 고객 지원

### AP

국가:	전화 번호
호주	1300-551-664 또는 03-9841-5211
중국	010-68002397
홍콩	2805-2563
인도네시아	+65 6100 6682
일본	+852 2805-2563
말레이시아	+65 6100 6682
뉴질랜드	09-574-2700
필리핀	+65 6100 6682
싱가포르	6100 6682
대한민국	2-561-2700
대만	+852 2805-2563
태국	+65 6100 6682
베트남	+65 6100 6682

### EMEA

국가:	전화 번호
오스트리아	01 360 277 1203
벨기에	02 620 00 86
벨기에	02 620 00 85
체코	296 335 612
덴마크	82 33 28 44
핀란드	09 8171 0281
프랑스	01 4993 9006
독일	069 9530 7103
그리스	210 969 6421
네덜란드	020 654 5301
아일랜드	01 605 0356
이탈리아	02 754 19 782
룩셈부르크	2730 2146

노르웨이	23500027
포르투갈	021 318 0093
러시아	495 228 3050
남아프리카 공화국	0800980410
스페인	913753382
스웨덴	08 5199 2065
스위스(프랑스어)	022 827 8780
스위스(독일어)	01 439 5358
스위스(이탈리아어)	022 567 5308
영국	0207 458 0161

**LA**

<b>국가:</b>	<b>전화 번호</b>
앵귈라	1-800-711-2884
앤티가	1-800-711-2884
아르헨티나	0-800- 555-5000
아루바	800-8000 ◆ 800-711-2884
바하마	1-800-711-2884
바베이도스	1-800-711-2884
버뮤다	1-800-711-2884
볼리비아	800-100-193
브라질	0-800-709-7751
영국령 버진 아일랜드	1-800-711-2884
케이맨 제도	1-800-711-2884
쿠라카오	001-800-872-2881 + 800-711-2884
칠레	800-360-999
콜롬비아	01-8000-51-4746-8368 (01-8000-51- HP INVENT)
코스타리카	0-800-011-0524
도미니카	1-800-711-2884
도미니카 공화국	1-800-711-2884



에콰도르	1-999-119 ◆ 800-711-2884(Andinatel) 1-800-225-528 ◆ 800-711-2884(Pacifitel)
엘살바도르	800-6160
프랑스령 앤틸리스	0-800-990-011 ◆ 800-711-2884
프랑스령 기아나	0-800-990-011 ◆ 800-711-2884
그레나다	1-800-711-2884
과달루페	0-800-990-011 ◆ 800-711-2884
과테말라	1-800-999-5105
가이아나	159 ◆ 800-711-2884
아이티	183 ◆ 800-711-2884
온두라스	800-0-123 ◆ 800-711-2884
자메이카	1-800-711-2884
마르티니크	0-800-990-011 ◆ 877-219-8671
멕시코	01-800-474-68368(800 HP INVENT)
몬트세라트	1-800-711-2884
네덜란드령 앤틸리스	001-800-872-2881 ◆ 800-711-2884
니카라과	1-800-0164 ◆ 800-711-2884
파나마	001-800-711-2884
파라과이	(009) 800-541-0006
페루	0-800-10111
푸에르토리코	1-877 232 0589
세인트 루시아	1-800-478-4602
세인트 빈센트	01-800-711-2884
세인트 크리스토퍼 네비스	1-800-711-2884
세인트 마틴	1-800-711-2884
수리남	156 ◆ 800-711-2884
트리니다드 토바고	1-800-711-2884
터크스 케이커스 제도	01-800-711-2884
미국령 버진 아일랜드	1-800-711-2884

우루과이	0004-054-177
베네수엘라	0-800-474-68368(0-800 HP INVENT)

**NA**

국가:	전화 번호
캐나다	800-HP-INVENT
미국	800-HP INVENT

최신 서비스 및 지원 정보를 보려면 <http://www.hp.com>에 로그인하십시오.

---

## Regulatory information

### Federal Communications Commission Notice

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio or television technician for help.

#### Modifications

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Hewlett-Packard Company may void the user's authority to operate the equipment.

#### Declaration of Conformity for Products Marked with FCC Logo, United States Only

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

If you have questions about the product that are not related to this declaration, write to

Hewlett-Packard Company  
P. O. Box 692000, Mail Stop 530113  
Houston, TX 77269-2000

For questions regarding this FCC declaration, write to  
Hewlett-Packard Company  
P. O. Box 692000, Mail Stop 510101

Houston, TX 77269-2000  
or call HP at 281-514-3333

To identify your product, refer to the part, series, or model number located on the product.

### Canadian Notice

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

### Avis Canadien

Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.



### European Union Regulatory Notice

This product complies with the following EU Directives:

- Low Voltage Directive 2006/95/EC
- EMC Directive 2004/108/EC

Compliance with these directives implies conformity to applicable harmonized European standards (European Norms) which are listed on the EU Declaration of Conformity issued by Hewlett-Packard for this product or product family.

This compliance is indicated by the following conformity marking placed on the product:

 <p>This marking is valid for non-Telecom products and EU harmonized Telecom products (e.g. Bluetooth).</p>	 <p>This marking is valid for EU non-harmonized Telecom products.</p> <p>*Notified body number (used only if applicable - refer to the product label)</p>
--	--

Hewlett-Packard GmbH, HQ-TRE, Herrenberger Strasse 140, 71034 Boeblingen, Germany

## Japanese Notice

この装置は、情報処理装置等電波障害自主規制協議会（VCCI）の基準に基づくクラスB情報技術装置です。この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に近接して使用されると、受信障害を引き起こすことがあります。

取扱説明書に従って正しい取り扱いをしてください。

## EU(유럽 연합) 가정용 사용자의 장비 폐기물 처리



제품 또는 포장지에 있는 이 표시는 해당 제품이 기타 가정 쓰레기와 같이 폐기해서는 안됨을 나타냅니다. 사용자는 전기/전기 장비 폐기물의 재활용을 위해 지정된 수거점에 장비 폐기물을 넘겨 주어야 할 책임이 있습니다. 폐기 시 장비 폐기물을 개별 수거하고 재활용하면 자연 자원을 보존하는 데 도움이 되고 이를 통해 신체 건강과 환경을 보호하는 방향으로 폐기물을 재활용할 수 있습니다.

재활용을 위해 장비 폐기물을 폐기할 수 있는 장소에 대한 자세한 내용은 지역 시청이나 가정 쓰레기 처리 서비스 부서나 또는 제품을 구매한 매장에 문의 바랍니다.

## Perchlorate Material - special handling may apply

This calculator's Memory Backup battery may contain perchlorate and may require special handling when recycled or disposed in California.



## 사용자 메모리 및 스택

---

이 부록에서 다루는 내용을 요약하면 다음과 같습니다.


- 사용자 메모리의 할당 및 요구 사항
- 메모리에 영향을 주지 않고 계산기를 재설정하는 방법
- 사용자 메모리를 지우고(삭제) 시스템 기본값을 재설정하는 방법
- 스택 올리기에 영향을 주는 연산

---

### 계산기 메모리 관리

HP 35s에는 모든 저장된 데이터 조합(변수, 방정식 또는 프로그램 줄)에 사용할 수 있는 30KB의 사용자 메모리가 있습니다. SOLVE, ∫FN 및 통계 계산에도 사용자 메모리가 필요합니다. ∫FN 연산을 실행할 때는 특히 "많은" 메모리가 필요합니다.

저장된 모든 데이터는 명시적으로 지우기 전까지는 유지됩니다. MEMORY FULL 메시지는 현재 메모리가 부족하여 방금 시도한 작업을 제대로 수행할 수 없음을 의미합니다. 이 경우에는 사용자 메모리의 일부 또는 전체를 지워야 합니다. 예를 들어 다음과 같은 조치를 취할 수 있습니다.

- 임의의 방정식 또는 모든 방정식을 지웁니다(6장의 "방정식 편집 및 지우기" 참조).
- 임의의 프로그램 또는 모든 프로그램을 지웁니다(13장의 "한 개 또는 여러 개의 프로그램 지우기" 참조).
- 모든 사용자 메모리를 지웁니다. 이를 위해서는  CLEAR **3** (3ALL)을 누릅니다.

사용 가능한 메모리 양을 확인하려면  MEM을 누릅니다. 디스플레이에 사용 가능한 바이트 수가 표시됩니다.

방정식 목록에서 특정 방정식의 메모리 요구 사항을 확인하려면

1. **[EQN]**을 눌러 방정식 모드를 활성화합니다. **EQN LIST TOP** 또는 현재 방정식의 왼쪽 끝이 표시됩니다.
2. 필요한 경우 **[^]** 또는 **[v]**를 눌러 원하는 방정식이 나타날 때까지 방정식 목록을 스크롤합니다.
3. **[F4] [SHOW]**를 눌러 방정식의 체크섬(16진) 및 길이(바이트)를 확인합니다. 예를 들어 **CK=382E LN=41**이 있습니다.

특정 프로그램의 전체 메모리 요구 사항을 확인하려면

1. **[F4] [MEM] [2]** (**2PGM**)를 눌러 프로그램 목록에서 첫 번째 레이블을 표시합니다.
2. 프로그램 목록을 스크롤합니다. 원하는 프로그램 레이블 및 크기가 나타날 때까지 **[^]** 또는 **[v]**를 누릅니다. 예를 들어 **LBL F LN=57**이 있습니다.
3. 옵션: **[F4] [SHOW]**를 눌러 프로그램의 체크섬(16진) 및 길이(바이트)를 확인합니다. 예를 들어 **CK=9CC9 LN=57**이 있습니다.

프로그램에서 방정식의 메모리 요구 사항을 확인하려면

1. 방정식이 포함된 프로그램 줄을 표시합니다.
2. **[F4] [SHOW]**를 눌러 체크섬 및 길이를 확인합니다. 예를 들어 **CK=AB71 LN=15**가 있습니다.



---

## 계산기 재설정

계산기가 키 입력에 반응하지 않거나 비정상적으로 작동하는 경우 재설정해 보십시오. 계산기를 재설정하면 현재 계산이 중단되고 프로그램 입력, 숫자 입력, 실행 중인 프로그램, SOLVE 계산, I/FN 계산, VIEW 디스플레이 또는 INPUT 디스플레이가 취소됩니다. 저장된 데이터는 일반적으로 그대로 유지됩니다.

계산기를 재설정하려면 **[C]** 키를 누른 채 **[GTO]**를 누릅니다. 계산기를 재설정할 수 없는 경우에는 새 배터리로 교체해 보십시오. 계산기를 재설정할 수 없거나 계속 제대로 작동하지 않는 경우 다음 절에서 설명하는 특별한 절차를 수행하여 메모리를 지우도록 해야 합니다.

계산기가 키 입력에 여전히 반응하지 않으면 가늘고 뾰족한 물체를 사용하여 RESET 구멍을 누릅니다.

계산기를 떨어뜨리거나 전원 공급이 중단된 경우 자체적으로 재설정될 수 있습니다.

---

## 메모리 지우기

사용자 메모리를 지우는 일반적인 방법은 **[MC]** **[CLEAR]** **[3]** (**3ALL**)을 누르는 것입니다. 그러나 추가 정보를 재설정하는 보다 강력한 지우기 절차도 있으며 이는 키 보드가 제대로 작동하지 않는 경우 유용합니다.

계산기가 키 입력에 반응하지 않고, 계산기를 재설정하거나 배터리를 교체해도 계산기가 다시 제대로 작동하지 않는 경우 다음과 같은 MEMORY CLEAR 절차를 수행해 보십시오. 이러한 키 입력은 모든 메모리를 지우고, 계산기를 재설정하고, 모든 형식 및 모드를 원래 상태인 *기본 설정*(아래에 표시)으로 복원합니다.

1. **[C]** 키를 누르고 있습니다.
2. **[R/S]**를 누르고 있습니다.
3. **[i]**를 누릅니다. 결국 이 세 개의 키를 동시에 누르게 됩니다. 세 개의 키를 모두 놓으면 계산기가 제대로 작동할 때 디스플레이에 MEMORY CLEAR가 표시됩니다.

범주	CLEAR ALL	MEMORY CLEAR (기본값)
각도 모드	변경되지 않음	도
진법 모드	변경되지 않음	10진수
대비 설정	변경되지 않음	보통
소수점	변경되지 않음	“.”
천 단위 구분 기호	변경되지 않음	“1,000”
분모(/c 값)	변경되지 않음	4095
디스플레이 형식	변경되지 않음	FIX 4
플래그	변경되지 않음	지워짐
복소수 모드	변경되지 않음	xiy
분수 디스플레이 모드	변경되지 않음	해제
난수 초기값	변경되지 않음	0
방정식 포인터	EQN LIST TOP	EQN LIST TOP
방정식 목록	지워짐	지워짐
FN = 레이블	Null	Null
프로그램 포인터	PRGM TOP	PRGM TOP
프로그램 메모리	지워짐	지워짐
스택 올리기	설정	설정
스택 레지스터	지워져서 0이 됨	지워져서 0이 됨
변수	지워져서 0이 됨	지워져서 0이 됨
간접 변수	정의되지 않음	정의되지 않음
논리	변경되지 않음	RPN

계산기를 떨어뜨리거나 전원 공급이 중단될 경우 메모리가 지워질 수 있습니다.

## 스택 올리지의 상태

4개의 스택 레지스터는 항상 존재하며 스택에는 항상 **스택 올리지 상태**가 있습니다. 다시 말해서 X 레지스터에 다음 숫자를 넣을 때 해당 동작과 관련하여 스택 올리지 가 항상 **설정**되거나 **해제**됩니다. 2장, "자동 메모리 스택"을 참조하십시오.

다음 두 목록에 있는 함수를 제외한 모든 함수의 경우 스택 올리지 가 설정됩니다.

### 해제 연산

**[ENTER]**, **[Σ+]**, **[Σ-]**, **[RCL]** **[CLEAR]** **[1]** (1 $\%$ ) 및 **[RCL]** **[CLEAR]** **[5]** (5 $\%$ STK)은 스택 올리지 를 해제합니다. 이러한 해제 연산 중 하나 뒤에 입력한 숫자가 현재 X 레지스터에 있는 숫자를 덮어씁니다. Y, Z 및 T 레지스터는 변경되지 않습니다.

또한 **[C]** 및 **[←]**가 CLx처럼 작동하면 스택 올리지 가 해제됩니다.

INPUT 함수는 프롬프트를 위해 프로그램을 중지할 때 스택 올리지 를 **해제**하여 입력한 숫자가 X 레지스터를 덮어쓰게 하지만 프로그램이 다시 시작하면 스택 올리지 를 **설정**합니다.

### 종립 연산

다음과 같은 연산은 스택 올리지 의 상태에 영향을 주지 않습니다.

DEG, RAD, GRAD	FIX, SCI, ENG, ALL	DEC, HEX, OCT, BIN	CLVARS
PSE	SHOW	RADIX . RADIX,	CLΣ
<b>[OFF]</b> <b>[RCL]</b> <b>[+]</b>	<b>[R/S]</b> 및 STOP	<b>[^]</b> 및 <b>[v]</b>	<b>[C]</b> * 및 <b>[←]</b> *
<b>[MEM]</b> <b>[1]</b> (1VAR)**	<b>[MEM]</b> <b>[2]</b> (2PGM)**	<b>[GTO]</b> <b>[.]</b> <b>[.]</b>	<b>[GTO]</b> <b>[.]</b> 레이블 nnn
EQN	FDISP	오류	<b>[PRGM]</b> 및 프로그램 입력
2진 창 전환	숫자 입력	xiy pθα	<b>[UNDO]</b>
* CLx처럼 사용되는 경우는 제외			
** {VAR} <b>[ENTER]</b> 및 {PGM} <b>[XEQ]</b> 를 제외하고 카탈로그가 표시되는 동안 수행되며 스택 올리지 를 설정하는 모든 연산 포함			

## LAST X 레지스터의 상태

다음 연산은 x를 RPN 모드의 LAST X 레지스터에 저장합니다.

+, -, ×, ÷	$\sqrt{x}$ , $x^2$ ,	$e^x$ , $10^x$
LN, LOG	$y^x$ , $\sqrt[y]{y}$	$1/x$ , INT±, Rmdr
SIN, COS, TAN	ASIN, ACOS, ATAN	↔ ↔
SINH, COSH, TANH	ASINH, ACOSH, ATANH	IP, FP, SGN, INTG, RND, ABS
%, %CHG	$\Sigma+$ , $\Sigma-$	RCL+, -, ×, ÷
	HMS→, →HMS	→DEG, →RAD
nCr nPr	!	ARG
CMLX +, -, ×, ÷	CMLX $e^x$ , LN, $y^x$ , $1/x$	CMLX SIN, COS, TAN
→kg, →lb	→°C, →°F	→cm, →in
→l, →gal	→KM →MILE	

↔는 LAST X 레지스터에 영향을 주지 않습니다.

다시 호출 산술 순서 **[X] [RCL] [+]** 변수는 x를 LASTx에 저장하고 **[X] [RCL]** 변수 **[+]**는 다시 호출한 숫자를 LASTx에 저장합니다.

ALG 모드에서 LAST X 레지스터는 스택의 부속 항목으로, 마지막 수식의 결과로 구한 숫자를 저장합니다. 이를 통해 ALG 모드에서 이전 수식의 결과를 사용할 수 있습니다.

---

## 스택 레지스터 내용 액세스

4개의 스택 레지스터 X, Y, Z 및 T에 저장된 값은 RPN 모드에서 REGX, REGY, REGZ 및 REGT 명령을 사용하여 방정식 또는 프로그램에서 액세스할 수 있습니다.

이러한 명령을 사용하려면 먼저 **[EQN]**을 누릅니다. 그리고 나서 **[R]**을 누르면 디스플레이에 X, Y, Z 및 T 레지스터를 표시하는 메뉴가 나타납니다. **[>]** 또는 **[<]**를 누르면 현재 레지스터를 선택했음을 나타내는 밑줄이 있는 기호가 이동합니다. **[ENTER]**를 누르면 선택된 스택 레지스터 값을 이후에 사용할 때 다시 호출하는 프로그램 또는 방정식에 명령이 수행됩니다. 이러한 명령은 REGX, REGY, REGZ 및 REGT로 표시됩니다.

예를 들어 먼저 **[EQN]**을 누른 다음 명령  $REGX \times REGY \times REGZ \times REGT$ 를 입력하여 프로그램 줄을 넣으면 4개의 스택 레지스터에서 값의 곱을 계산하고 그 결과를 X 레지스터에 넣습니다. 이렇게 하면 스택 레지스터 Y, Z 및 T에 이전 값인 X, Y 및 Z가 남습니다.

이렇게 스택에서 값을 효율적으로 사용하는 기능은 이러한 방식으로만 가능하며, HP35s에서는 다른 경우 이러한 기능을 지원하지 않습니다.



# ALG: 요약

## ALG 정보

이 부록에는 다음과 같이 ALG 모드와 관련된 일부 기능이 요약되어 있습니다.

- 두 인수 산술
- 지수 및 로그 함수 (  $\leftarrow$   $10^x$ ,  $\leftarrow$  LOG,  $\rightarrow$   $e^x$ ,  $\rightarrow$  LN )
- 삼각 함수
- 숫자의 특정 부분
- 스택 검토
- 복소수 연산
- 방정식 적분
- 2, 8 및 16진수의 산술
- 2개 변수 통계 데이터 입력

**MODE** **4**(4ALG)를 눌러 계산기를 ALG 모드로 설정합니다. 계산기가 ALG 모드에 있으면 ALG 표시가 켜집니다.

ALG 모드에서는 연산이 다음 순서대로 수행됩니다.

1. 괄호 안의 수식
2. 계승(!) 함수의 경우 **!**를 누르기 전에 값을 입력해야 합니다.
3. 함수 키를 누르기 전에 값을 입력해야 하는 함수에는 COS, SIN, TAN, ACOS, ASIN, ATAN, LOG, LN,  $x^2$ ,  $1/x$ ,  $\sqrt{x}$ ,  $\pi$ ,  $\sqrt[3]{x}$ , %, RND, RAND, IP, FP, INTG, SGN, nPr, nCr, %CHG, INT $\pm$ , Rmdr, ABS,  $e^x$ ,  $10^x$ , 단위 변환 등이 있습니다.
4.  $\sqrt[y]{x}$  및  $y^x$

5. 음수 +, -
6.  $\times$ ,  $\div$
7. +, -
8. =

## ALG에서 두 인수 산술 실행

ALG를 사용한 산술에 대한 본 설명은 ALG 모드의 영향을 받는 다음 부분을 대체합니다. 두 인수 산술 연산은 ALG 모드의 영향을 받습니다.

- 단순 산술
- 거듭제곱 함수( $y^x$ ,  $\sqrt[y]{x}$ )
- 백분율 계산(% 또는  $\rightarrow$  %CHG)
- 순열 및 조합( $\leftarrow$  nCr,  $\rightarrow$  nPr)
- 나눗셈의 몫 및 나머지( $\leftarrow$  INTG 2)(2INTG+), ( $\leftarrow$  INTG 3)(3Rmdr)

### 단순 산술

단순 산술의 예를 들면 다음과 같습니다.

ALG 모드에서 첫 번째 숫자를 입력하고, 연산자(+, -,  $\times$ ,  $\div$ )를 누르고, 두 번째 숫자를 입력하고, 마지막으로  $\rightarrow$  키를 누릅니다.

계산:	키:	디스플레이:
12 + 3	$\boxed{1} \boxed{2} \boxed{+} \boxed{3} \rightarrow$	12+3 15.0000
12 - 3	$\boxed{1} \boxed{2} \boxed{-} \boxed{3} \rightarrow$	12-3 9.0000
12 $\times$ 3	$\boxed{1} \boxed{2} \boxed{\times} \boxed{3} \rightarrow$	12 $\times$ 3 36.0000
12 $\div$ 3	$\boxed{1} \boxed{2} \boxed{\div} \boxed{3} \rightarrow$	12 $\div$ 3 4.0000



## 거듭제곱 함수

ALG 모드에서 숫자  $y$ 의  $x$ 제곱을 구하려면  $y$   $y^x$   $x$ 를 입력한 다음  $\text{[ENTER]}$ 를 누릅니다.

<b>계산:</b>	<b>키:</b>	<b>디스플레이:</b>
$12^3$	$\text{[1] [2] } y^x \text{ [3] [ENTER]}$	$12^3$ $1,728.0000$
$64^{1/3}$ (세제곱근)	$\text{[6] } x\sqrt{y} \text{ [3] } \text{[>] [6] [4] [ENTER]}$	$X\text{ROOT}(3,64)$ $4.0000$

## 백분을 계산

백분을 함수를 사용합니다.  $\text{[%]}$  키는 숫자를 100으로 나눕니다.

<b>계산:</b>	<b>키</b>	<b>디스플레이</b>
200의 27%	$\text{[2] [0] [%] [2] [0] [0] } \text{[>] [2] [7] [ENTER]}$	$\%(200,27)$ $54.0000$
200 - 27%	$\text{[2] [0] [0] [-] [2] [%] [2] [0] [0] } \text{[>] [2] [7] [ENTER]}$	$200 - \%(200,27)$ $146.0000$
25 + 12%	$\text{[2] [5] [+]} \text{[2] [%] [2] [5] } \text{[>] [1] [2] [ENTER]}$	$25 + \%(25,12)$ $28.0000$

계산:	키
$y$ 의 $x\%$	$\text{[2] [%] } y \text{ [>] } x \text{ [ENTER]}$
$y$ 에서 $x$ 로의 변동률. ( $y \neq 0$ )	$\text{[6] [%CHG] } y \text{ [>] } x \text{ [ENTER]}$

**예:**

15.76달러 항목의 비용이 작년에 16.12달러였다고 가정할 경우 작년 가격에서 금년 가격으로의 변동률은 얼마입니까?

<b>키</b>	<b>디스플레이</b>	<b>설명</b>
$\text{[6] [%CHG] [1] [6] [.] [1] [2] } \text{[>] [1] [5] [.] [7] [6] [ENTER]}$	$\%CHG(16.12,15.7$ $-2.2333$	금년 가격이 작년 가격보다 약 2.2% 인하되었습니다.

## 순열 및 조합

### 예: 직원 조합

14명의 여자 직원과 10명의 남자 직원을 고용하고 있는 회사에서 6인으로 이루어진 안전 위원회를 구성하려고 합니다. 만들 수 있는 직원 조합 수는 몇 개입니까

키	디스플레이	설명
<b>↵</b> <b>nCr</b> <b>2</b> <b>4</b> <b>&gt;</b> <b>6</b> <b>ENTER</b>	nCr(24,6) 134,596.0000	가능한 전체 조합 수입니다.

## 나눗셈의 몫과 나머지

**↵** **INTG** **2** (**2INTG+**) 및 **↵** **INTG** **3** (**3Rmdr**)을 사용하면 두 정수를 사용한 나눗셈 연산의 몫 또는 나머지를 구할 수 있습니다.

**↵** **INTG** **2** (**2INTG+**) 정수 1 **>** 정수 2 **ENTER**

**↵** **INTG** **3** (**3Rmdr**) 정수 1 **>** 정수 2 **ENTER**

예:

58 ÷ 9의 몫과 나머지를 표시합니다.

키	디스플레이	설명
<b>↵</b> <b>INTG</b> <b>2</b> ( <b>2INTG+</b> ) <b>5</b> <b>8</b> <b>&gt;</b> <b>9</b> <b>ENTER</b>	IDIV(58,9) 6.0000	몫을 표시합니다.
<b>↵</b> <b>INTG</b> <b>3</b> ( <b>3Rmdr</b> ) <b>5</b> <b>8</b> <b>&gt;</b> <b>9</b> <b>ENTER</b>	RMDR(58,9) 4.0000	나머지를 표시합니다.

## 괄호 계산

다른 숫자를 입력할 때까지 중간 결과 계산을 보류하려면 괄호를 사용합니다. 예를 들어, 다음을 계산하려고 합니다.

$$\frac{30}{85-12} \times 9$$

**3** **0** **÷** **8** **5** **-** **1** **2** **×** **9**를 입력하면 결과로 -107.6471이 나옵니다. 그러나 이 결과는 원하는 값이 아닙니다. 85-12를 먼저 계산한 다음 나눗셈을 계산하려면 괄호를 사용합니다.

키	디스플레이	설명
<b>3</b> <b>0</b> <b>÷</b> <b>(</b> <b>8</b> <b>5</b> <b>-</b>	30÷(85-)	계산이 수행되지 않습니다.
<b>1</b> <b>2</b> <b>)</b>	30÷(85-12) _	85 - 12를 계산합니다.
<b>×</b> <b>9</b>	30÷(85-12)×9 _	30/73을 계산합니다.
<b>ENTER</b>	30÷(85-12)×9 3.6986	30/(85 - 12) × 9를 계산합니다.

왼쪽 괄호 앞에 곱셈 부호(×)를 생략할 수 있습니다. 방정식 모드에서는 암시적 곱셈을 수행할 수 없습니다. 예를 들어, 수식 × (5 - 4)는 2와 왼쪽 괄호 사이에 **×** 키를 넣지 않고 **2** **(** **5** **-** **4**로 입력할 수 있습니다.

## 지수 및 로그 함수

계산:	키:	디스플레이:
자연 로그(밑 e)	<b>LN</b> <b>1</b> <b>ENTER</b>	LN(1) 0.0000
상용 로그(밑 10)	<b>LOG</b> <b>1</b> <b>0</b> <b>ENTER</b>	LOG(10) 1.0000
자연 지수	<b>e<sup>x</sup></b> <b>2</b> <b>ENTER</b>	EXP(2) 7.3891
상용 지수(역대수)	<b>10<sup>x</sup></b> <b>2</b> <b>ENTER</b>	ALOG(2) 100.0000

# 삼각 함수

각 단위를 **MODE** **1**(1DEG)로 가정합니다.

계산:	키:	디스플레이:
x의 사인	<b>SIN</b> <b>3</b> <b>0</b> <b>ENTER</b>	SIN(30) 0.5000
x의 코사인	<b>COS</b> <b>6</b> <b>0</b> <b>ENTER</b>	COS(60) 0.5000
x의 탄젠트	<b>TAN</b> <b>4</b> <b>5</b> <b>ENTER</b>	TAN(45) 1.0000
x의 아크사인	<b>ASIN</b> <b>1</b> <b>ENTER</b>	ASIN(1) 90.0000
x의 아크코사인	<b>ACOS</b> <b>0</b> <b>ENTER</b>	ACOS(0) 90.0000
x의 아크탄젠트	<b>ATAN</b> <b>0</b> <b>ENTER</b>	ATAN(0) 0.0000

# 쌍곡선 함수

계산:	키:
x의 쌍곡선 사인(SINH)	<b>HYP</b> <b>SIN</b> , 숫자 입력, <b>ENTER</b> 누르기
x의 쌍곡선 코사인(COSH)	<b>HYP</b> <b>COS</b> , 숫자 입력, <b>ENTER</b> 누르기
x의 쌍곡선 탄젠트(TANH)	<b>HYP</b> <b>TAN</b> , 숫자 입력, <b>ENTER</b> 누르기
x의 쌍곡선 아크사인(ASINH)	<b>HYP</b> <b>ASIN</b> , 키 입력, <b>ENTER</b> 누르기
x의 쌍곡선 아크코사인(ACOSH)	<b>HYP</b> <b>ACOS</b> , 키 입력, <b>ENTER</b> 누르기
x의 쌍곡선 아크탄젠트(ATANH)	<b>HYP</b> <b>ATAN</b> , 키 입력, <b>ENTER</b> 누르기

## 숫자의 특정 부분

계산:	키:	디스플레이:
2.47의 정수부	$\left[ \leftarrow \right]$ INTG $\left[ 6 \right]$ (6IP) $\left[ 2 \right]$ $\left[ \cdot \right]$ $\left[ 4 \right]$ $\left[ 7 \right]$ ENTER	IP(2.47) 2.0000
2.47의 소수부	$\left[ \leftarrow \right]$ INTG $\left[ 5 \right]$ (5FP) $\left[ 2 \right]$ $\left[ \cdot \right]$ $\left[ 4 \right]$ $\left[ 7 \right]$ ENTER	FP(2.47) 0.4700
7의 절대값	$\left[ \rightarrow \right]$ ABS $\left[ +\leftarrow \right]$ $\left[ 7 \right]$ ENTER	ABS(-7) 7.0000
9의 부호 값	$\left[ \leftarrow \right]$ INTG $\left[ 1 \right]$ (1SGN) $\left[ 9 \right]$ ENTER	SGN(9) 1.0000
5.3보다 작거나 같은 가장 큰 정수	$\left[ \leftarrow \right]$ INTG $\left[ 4 \right]$ (4INTG) $\left[ +\leftarrow \right]$ $\left[ 5 \right]$ $\left[ \cdot \right]$ $\left[ 3 \right]$ ENTER	INTG(-5.3) -6.0000

## 스택 검토

$\left[ \uparrow \right]$  또는  $\left[ \rightarrow \right]$   $\left[ \uparrow \right]$  키를 누르면 디스플레이에 X, Y, Z, T 레지스터 메뉴가 표시되므로 스택의 전체 내용을 검토할 수 있습니다.  $\left[ \uparrow \right]$  키와  $\left[ \rightarrow \right]$   $\left[ \uparrow \right]$  키의 차이점은 디스플레이에서의 밑줄 위치입니다.  $\left[ \rightarrow \right]$   $\left[ \uparrow \right]$ 을 누르면 T 레지스터에 밑줄이 표시되고,  $\left[ \uparrow \right]$ 을 누르면 Y 레지스터에 밑줄이 표시됩니다.

$\left[ \uparrow \right]$ 을 누르면 다음 메뉴가 표시됩니다.

X Y Z T

Z —

$\left[ \rightarrow \right]$   $\left[ \uparrow \right]$ 을 누르면 다음 메뉴가 표시됩니다.

X Y Z T

Z —

$\left[ \uparrow \right]$  및  $\left[ \rightarrow \right]$   $\left[ \uparrow \right]$ 을  $\left[ \rightarrow \right]$  또는  $\left[ \leftarrow \right]$ 와 함께 누르면 스택의 전체 내용을 검토하고 다시 호출할 수 있습니다. 이는 다시 호출한 누적 부분 및 이후 계산에 이러한 누적 부분을 사용할 수 있는지 여부에 따라 REGX, REGY, REGZ 또는 REGT로 표시됩니다.

ALG 모드에서의 X, Y, Z, T 레지스터 값은 RPN 모드에서와 동일합니다. 일반 계산, 풀이, 프로그래밍 또는 적분 이후 4개의 레지스터 값은 RPN 또는 ALG 모드에서 서로 동일하고 ALG 논리 모드와 RPN 논리 모드 사이를 전환할 때 유지됩니다.

---

## 방정식 적분

1. 방정식을 입력하고(6장의 "방정식 목록에 방정식 입력" 참조) 방정식 모드를 종료합니다.
2. 적분 한계를 입력합니다. *하한값*을 입력하고  $[x \leftrightarrow y]$ 를 누른 다음 상한값을 입력합니다.
3. 방정식을 표시합니다.  $[EQN]$ 을 누르고, 필요한 경우  $[\wedge]$  또는  $[v]$ 를 눌러 방정식 목록을 스크롤하여 원하는 방정식을 표시합니다.
4. 적분 변수를 선택합니다.  $[<]$   $[/]$  *변수를 누릅니다*. 계산이 시작됩니다.

---

## 복소수 연산

복소수를 입력하려면:

형태 :  $x \cdot i^y$

1. 실수부를 입력합니다.
2.  $[i]$ 를 누릅니다.
3. *허수부*를 입력합니다.

형태 :  $x + y \cdot i$

1. 실수부를 입력합니다.
2.  $[+]$ 를 누릅니다.
3. *허수부*를 입력합니다.
4.  $[i]$ 를 누릅니다.

형태 :  $r \theta a$

1.  $r$ 의 값을 입력합니다.
2.  $[r]$   $[\theta]$ 를 누릅니다.
3.  $\theta$ 의 *값*을 입력합니다.

### 하나의 복소수로 연산을 수행하려면

1. 함수를 선택합니다.
2. 복소수  $z$ 를 입력합니다.
3. **[ENTER]**를 눌러 계산합니다.
4. 계산 결과가 줄 2에 표시되고, 표시된 형태는 **[MODE]**에서 설정한 형태입니다.

### 두 개의 복소수로 산술 연산을 수행하려면

1. 첫 번째 복소수  $z_1$ 을 입력합니다.
2. 산술 연산을 선택합니다.
3. 두 번째 복소수  $z_2$ 를 입력합니다.
4. **[ENTER]**를 눌러 계산합니다.
5. 계산 결과가 줄 2에 표시되고, 표시된 형태는 **[MODE]**에서 설정한 형태입니다.

복소수의 예를 들면 다음과 같습니다.

**예:**

$\sin(2 + 3i)$ 을 계산합니다.

키:	디스플레이:	설명:
<b>[<math>\leftarrow</math>]</b> <b>[DISPLAY]</b> <b>[9]</b> ( $9 \times i$ )		디스플레이 형식을 설정합니다.
<b>[SIN]</b> <b>[2]</b> <b>[+]</b> <b>[3]</b> <b>[i]</b>	$\text{SIN}(2+3 \cdot i)$	
<b>[ENTER]</b>	$\text{SIN}(2+3i)$	결과는
	$9.1545i - 4.1689$	$9.1545 i - 4.1689$ 입니다.

**예:**

다음 수식을 계산합니다.

$$z_1 \div (z_2 + z_3),$$

여기서  $z_1 = 23 + 13i$ ;  $z_2 = 2 + i$ ;  $z_3 = 4 - 3i$ 입니다.

키:

[←] [DISPLAY] [.] [3] [i]  
 (10x+y·i)  
 [()]  
 [2] [3] [+ ] [1] [3] [i]  
 [>] [÷] [()] [+/-] [2] [+]  
 [i] [+ ] [4] [- ] [3] [i]  
 [ENTER]

디스플레이:

←i ÷ (-2+i+4-3i)  
 (23+13i) ÷ (-2+...  
 2.5000+9.0000i

설명:

디스플레이 형식을 설정합니다.

결과는 2.5000 + 9.0000 i입니다.

예:

Evaluate  $(4 - 2/5 i) \times (3 - 2/3 i)$  를 계산합니다.

키:

[()] [4] [- ] [.] [2] [.]  
 [5] [i] [ > ] [x] [()] [3]  
 [- ] [.] [2] [.] [3] [i]  
 [ENTER]

디스플레이:

←5i) × (3-0 2/3i)  
 (4-0 2/5i) × (3...  
 11.7333i-3.8667

설명:

결과는 11.7333 i-3.8667입니다.

## 2, 8 및 16진수의 산술

16진, 8진 및 2진 모드의 산술 예는 다음과 같습니다.

예:

$$12F_{16} + E9A_{16} = ?$$

키:

[→] [BASE] [2] (2HEX)  
 [1] [2] [RCL] [F] [→]  
 [BASE] [6] (6h) [+ ] [RCL]  
 [E] [9] [RCL] [A] [→]  
 [BASE] [6] (6h) [ENTER]

디스플레이:

12Fh+E9Ah  
 FC9h

설명:

16진수를 설정하고 HEX 표시기가 켜집니다. 결과입니다.



$$7760_g - 4326_g = ?$$

**[F]** **[BASE]** **[3]** (3OCT)

$$12Fh + E9Ah$$

8진수를 설정하고 OCT

$$77110$$

표시기가 켜집니다.

**[7]** **[7]** **[6]** **[0]** **[F]** **[BASE]**

$$7760o - 4326o$$

표시된 숫자를 8진수로

**[7]** (7o)

$$3432o$$

변환합니다.

**[-]** **[4]** **[3]** **[2]** **[6]** **[F]**

**[BASE]** **[7]** (7o) **[ENTER]**

$$100_g \div 5_g =$$

**[1]** **[0]** **[0]** **[F]** **[BASE]** **[7]**

$$100o \div 5o$$

결과의 정수부입니다.

(7o) **[÷]** **[5]** **[F]**

$$14o$$

**[BASE]** **[7]** (7o) **[ENTER]**

$$5A0_{16} + 10011000_2 =$$

**[F]** **[BASE]** **[2]** (2HEX)

$$5A0h +$$

16진수를 설정하고 HEX

**[5]** **[RCL]** **[A]** **[0]** **[F]**

표시기가 켜집니다.

**[BASE]** **[6]** (6h) **[+]**

**[1]** **[0]** **[0]** **[1]** **[1]** **[0]** **[0]** **[←]**  $A0h + 10011000b$

**[0]** **[F]** **[BASE]** **[8]** (8b)

$$5A0h + 10011000b$$

16진수로 표시된 결과입

**[ENTER]**

$$638h$$

니다.

**[F]** **[BASE]** **[1]** (1DEC)

$$5A0h + b10011000b$$

10진수로 복원합니다.

$$1,592.0000$$

## 2개 변수 통계 데이터 입력

ALG 모드에서는  $y$ 가 Y 레지스터에 오고,  $X$ 가 X 레지스터에 오도록 ( $x, y$ ) 쌍을 역순( $y$  **[x↔y]**  $x$  또는  $y$  **[ENTER]**  $x$ )으로 입력해야 합니다.

- [F]** **[CLEAR]** **[4]** (4Σ)를 눌러 기존 통계 데이터를 지웁니다.
- 먼저  $y$  값을 입력하고 **[x↔y]**를 누릅니다.
- 해당  $x$  값을 입력하고 **[Σ+]**를 누릅니다.
- 디스플레이에 누적된 통계 데이터 쌍의 개수인  $n$ 이 표시됩니다.
- $x, y$  쌍의 입력을 계속합니다.  $n$ 이 각 입력으로 업데이트됩니다.

잘못된 값이 방금 입력한 값인 경우  $\leftarrow$   $\Sigma^-$ 를 누릅니다. 잘못된 통계 데이터를 삭제하면 계산기에서는 입력한 마지막 통계 데이터를 줄 1(디스플레이의 맨 윗줄)에 표시하고,  $n=0$ 을 줄 2에 표시합니다.

**예:**

왼쪽에  $x, y$  값을 입력한 다음 오른쪽에 표시된 대로 수정합니다.

초기 $x, y$	수정된 $x, y$
20, 4	20, 5
400, 6	40, 6

키:	디스플레이:	설명:
$\rightarrow$ CLEAR 4 (4 $\Sigma$ )	20 $\Sigma^+$	기존 통계 데이터를 지웁니다.
4 $x \leftrightarrow y$ 2 0 $\Sigma^+$	1.0000	첫 번째 새 데이터 쌍을 입력합니다.
6 $x \leftrightarrow y$ 4 0 0	400 $\Sigma^+$	디스플레이에 입력된 데이터 쌍의
$\Sigma^+$	2.0000	개수인 $n$ 이 표시됩니다.
$\rightarrow$ LAST $x$	LAST $x$	마지막 $x$ 값을 다시 호출합니다.
	400.0000	마지막 $y$ 는 여전히 Y 레지스터에
		있습니다.
$\leftarrow$ $\Sigma^-$	400 $\Sigma^-$	마지막 데이터 쌍을 삭제합니다.
	1.0000	
6 $x \leftrightarrow y$ 4 0 $\Sigma^+$	40 $\Sigma^+$	마지막 데이터 쌍을 다시 입력합니
	2.0000	다.
4 $x \leftrightarrow y$ 2 0 $\leftarrow$	20 $\Sigma^-$	첫 번째 데이터 쌍을 삭제합니다.
$\Sigma^-$	1.0000	
5 $x \leftrightarrow y$ 2 0 $\Sigma^+$	20 $\Sigma^+$	첫 번째 데이터 쌍을 다시 입력합니
	2.0000	다. 여전히 통계 레지스터에 두 데이
		터 쌍의 합계가 있습니다.

## 선형 회귀

선형 회귀인 L.R.은 *선형 추산*이라고도 하며,  $x, y$  데이터 세트에 가장 맞는 직선을 구하는 통계 방법입니다.

- $x$  또는  $y$ 에 대한 추산 값을 구하려 제공된  $y$  또는  $x$ 의 가상 값을 입력하고 **ENTER**를 누른 다음 **↩** **L.R.** ( $\hat{x}$ ) 또는 **↩** **L.R.** **>** ( $\hat{y}$ )를 누릅니다.
- 데이터에 가장 맞는 선을 정의하는 값을 구하려면 **↩** **L.R.** 뒤에 **(r)**, **(m)** 또는 **(b)**를 누릅니다.



## 방정식 풀기 추가 정보

이 부록에서는 7장에 설명된 내용 이외의 SOLVE 연산 관련 정보를 제공합니다.

### SOLVE에서 근을 구하는 방법

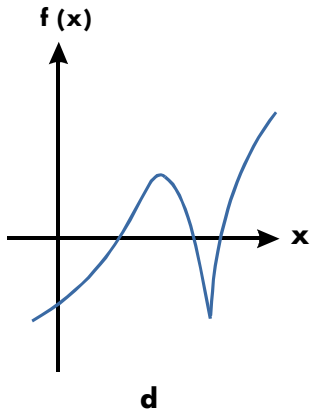
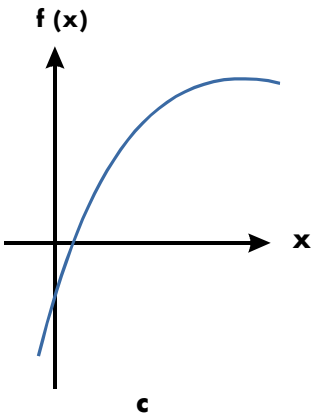
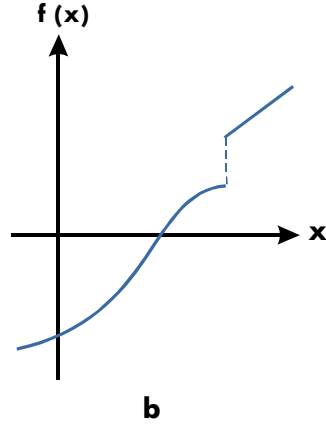
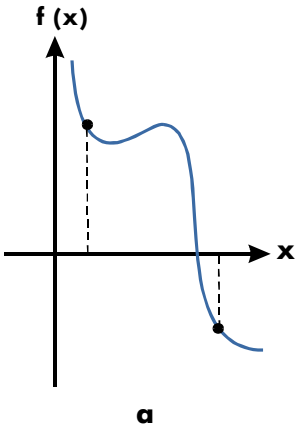
SOLVE는 먼저 방정식에서 직접 미지수 변수 값을 구하려고 합니다. 이 시도가 실패하면 SOLVE는 반복 절차를 수행합니다. *반복* 연산은 지정된 방정식을 반복 실행하는 것입니다. 방정식에서 구한 값은 미지수 변수  $x$ 의  $f(x)$  함수입니다.  $f(x)$ 는 미지수 변수  $x$ 에 대해 정의된 함수의 수학적 표기입니다. SOLVE는 먼저 미지수 변수  $x$ 에 대한 추정값으로 시작하고,  $f(x)$  함수를 각각 연속 실행하여 해당 추정값을 구체화합니다.

$f(x)$  함수에 대한 연속 두 추정값의 부호가 서로 반대인 경우 SOLVE에서는  $f(x)$  함수가 두 추정값 사이의 적어도 한 위치에서  $x$  축을 교차한다고 가정합니다. 이러한 간격은 근을 찾을 때까지 체계적으로 좁혀져 있습니다.

SOLVE에서 근을 찾으려면 근이 계산기의 숫자 범위 내에 있어야 하고, 반복적으로 값을 검색하는 위치에 함수가 수학적으로 정의되어 있어야 합니다. 다음과 같은 조건 중 하나 이상을 충족하는 경우 SOLVE에서는 오버플로 경계 내에 있는 근을 찾습니다.

- 두 추정값으로 인해 서로 부호가 반대인  $f(x)$  값이 구해지고, 함수 그래프에서 이러한 두 추정값 사이의 적어도 한 위치에서  $x$  축을 교차시킵니다(아래의 그림 a 참조).
- $x$ 가 늘어날 때  $f(x)$ 가 항상 늘어나거나 줄어듭니다(아래의 그림 b 참조).
- $f(x)$  그래프는 모든 위치에서 오목하거나 볼록합니다(아래의 그림 c 참조).

- $f(x)$ 의 지역 최소값 또는 최대값이 하나 이상 있는 경우 각각의 값은  $f(x)$ 의 인접 근 사이에 하나씩 놓입니다(아래의 그림 d 참조).



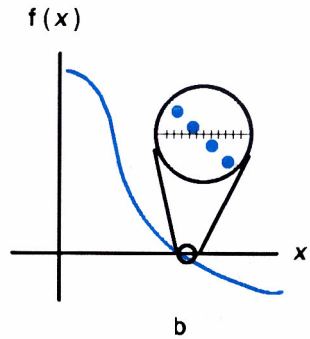
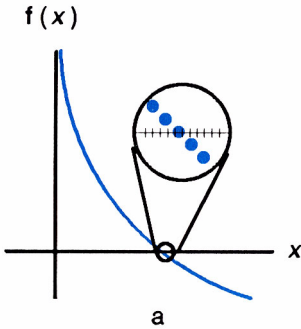
**Function Whose Roots Can Be Found**

대부분의 경우 계산된 근은 방정식의 이론적으로 매우 정확한 근을 정확하게 추정한 값입니다. "이상적인" 해는  $f(x) = 0$ 에 대한 값입니다. 그러나 12자리의 전체 자릿수 제한이 있는 숫자의 근사값 계산 결과로 나올 수 있는  $f(x)$ 에 대한 매우 작은 0이 아닌 값을 받아들여야 하는 경우도 있습니다.

## 결과 해석

SOLVE 연산은 다음과 같은 조건 중 하나에서 해를 구합니다.

- $f(x)$ 가 0인 추정값을 구하는 경우. 아래의 그림 a를 참조하십시오.
- $f(x)$ 가 0은 아니지만 계산된 근이 함수 그래프가  $x$  축을 교차하는 곳에 인접한 12자리 숫자인 경우(아래의 그림 b 참조). 이는 두 개의 최종 추정값이 서로 인접해 있고(서로 12번째 자릿수에서 1만큼 떨어져 있음) 함수 값이 한 추정값에 대해서는 양수이고 다른 추정값에 대해서는 음수인 경우 가능합니다. 그렇지 않으면 이러한 값은 (0, 10-499) 또는 (0, -10-499)입니다. 대부분의 경우에는  $f(x)$ 가 0에 매우 가깝습니다.



- ✓ 결과에 대한 자세한 내용을 보려면 **[R↓]**을 눌러 Y 레지스터에 남은 근( $x$ )의 이전 추정값을 봅니다. **[R↓]**을 다시 눌러 Z 레지스터에 남은  $f(x)$ 의 값을 봅니다.  $f(x)$ 가 0이거나 매우 작은 경우 해를 구했을 가능성이 매우 높습니다. 그러나  $f(x)$ 가 매우 큰 경우 결과를 해석할 때 주의해야 합니다.

### 예: 근이 하나인 방정식

방정식의 근을 구합니다.

$$-2x^3 + 4x^2 - 6x + 8 = 0$$

방정식을 수식으로 입력합니다.

키:

[EQN]  
 [+/-] [2] [X]  
 [RCL] [X] [y<sup>x</sup>] [3]  
 [+] [4] [X]  
 [RCL] [X] [y<sup>x</sup>] [2]  
 [-] [6] [X] [RCL] [X]  
 [+] [8] [ENTER]  
 [↵] [SHOW]

[C]

이제 방정식을 풀어 근을 구합니다.

키:

[0] [↵] [STO] [X]  
 [ENTER] [1] [0]  
 [EQN]

[↵] [SOLVE] [X]

✓ [R↓]

✓ [R↓]

디스플레이:

$-2x^3+4x^2-6$  →

CK=B99D

LN=18

10\_

$-2x^3+4x^2-6$  →

SOLVING

X=

1.6506

1.6506

$-4.00000E-11$

설명:

방정식 모드를 선택합니다.

방정식을 입력합니다.

체크섬 및 길이입니다.

방정식 모드를 취소합니다.

설명:

근의 초기 추정값입니다.

방정식 모드를 선택하고 방정식의 왼쪽 끝을 표시합니다.

X의 값을 구하고 결과를 표시합니다.

두 최종 추정값의 소수 네 자릿수가 서로 동일합니다.

$f(x)$ 가 매우 작으므로 근사값이 적합한 근입니다.

## 예: 근이 두 개인 방정식

포물선 방정식의 두 근을 구합니다.

$$x^2 + x - 6 = 0.$$

방정식을 수식으로 입력합니다.



키:	디스플레이:	설명:
<b>[EQN]</b>		방정식 모드를 선택합니다.
<b>[RCL] [X] <math>y^x</math> [2] [+]</b>		방정식을 입력합니다.
<b>[RCL] [X] [-] [6]</b>	$X^2+X-6$	
<b>[ENTER]</b>		
<b>[↵] [SHOW]</b>	CK=3971 LN=7	체크섬 및 길이입니다.
<b>[C]</b>		방정식 모드를 취소합니다.

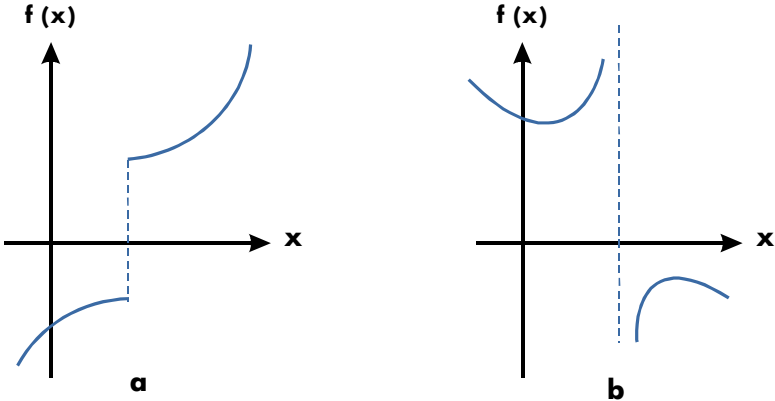
이제 방정식을 풀어 양수 근 및 음수 근을 구합니다.

키:	디스플레이:	설명:
<b>[0] [↵] [STO] [X]</b>	10_	양수 근의 초기 추정값입니다.
<b>[ENTER] [1] [0]</b>		
<b>[EQN]</b>	$X^2+X-6$	방정식 모드를 선택하고 방정식을 표시합니다.
<b>[↵] [SOLVE] [X]</b>	SOLVING X= 2.0000	추정값 0 및 10을 사용하여 양수 근을 계산합니다.
<b>[R↓]</b>	2.0000	두 최종 추정값이 서로 동일합니다.
<b>[R↓] [↵] [SHOW]</b>	0.000000000000	$f(x) = 0$
<b>[0] [↵] [STO] [X]</b>	-10_	음수 근의 초기 추정값입니다.
<b>[ENTER] [1] [0] [+/-]</b>		
<b>[EQN]</b>	$X^2+X-6$	방정식을 다시 표시합니다.
<b>[↵] [SOLVE] [X]</b>	SOLVING X= -3.0000	추정값 0 및 -10을 사용하여 음수 근을 계산합니다.
<b>[R↓] [R↓] [↵] [SHOW]</b>	0.000000000000	$f(x) = 0$

특별히 신경써야 할 경우가 있습니다.

- $x$  축이 교차하는 곳에서 함수의 그래프가 끊어진 경우 SOLVE 연산에서 이렇게 끊어진 곳에 인접한 값을 반환합니다(아래의 그림 a 참조). 이 경우  $f(x)$ 가 매우 클 수 있습니다.

- $f(x)$  값은 그래프에서 부호가 바뀌는 위치에서 무한대에 이를 수 있습니다 (아래의 그림 b 참조). 이러한 경우를 극이라고 합니다. SOLVE 연산에서  $x$ 의 두 인접 값 사이에 부호가 변경되었음을 확인하기 때문에 가능한 근을 반환합니다. 그러나  $f(x)$  값이 매우 큼니다. 정확하게 12자리로 표시된  $x$  값에서 극이 생긴 경우에는 해당 값으로 인해 계산이 중단되고 오류 메시지가 나타납니다.



**Special Case: A Discontinuity and a Pole**

**예: 불연속 함수**

방정식의 근을 구합니다.

$$IP(x) = 1.5$$

방정식을 입력합니다.

**키:**

**디스플레이:**

**설명:**

EQN  
 [←] [INTG] [6] (6IP)  
 RCL [X] [>] [←] [=]  
 1 [.] [5] [ENTER]  
 [←] [SHOW]  
 [C]

IP(X)=1.5  
 CK=D2C1  
 LN=9

방정식 모드를 선택합니다.  
 방정식을 입력합니다.

체크섬 및 길이입니다.

방정식 모드를 취소합니다.

이제 방정식을 풀어 근을 구합니다.

키:	디스플레이:	설명:
0 $\rightarrow$ STO $\rightarrow$ X	5_	근의 초기 추정값입니다.
ENTER 5	IP(X)=1.5	방정식 모드를 선택하고 방정식을 표시합니다.
EQN	SOLVING	추정값 0 및 5를 사용하여 근을 구합니다.
$\rightarrow$ SOLVE X	X=	
	2.0000	
$\leftarrow$ SHOW	1.999999999999	소수 11자릿수의 근을 표시합니다.
$\leftarrow$ $\leftarrow$ SHOW	2.000000000000	이전 추정값이 약간 더 큼니다.
$\leftarrow$ $\leftarrow$ $\leftarrow$ SHOW	-0.5000	$f(x)$ 이 매우 큼니다.

두 최종 추정값이 서로 다르고  $f(x)$  값이 매우 큼을 확인할 수 있습니다. 문제는  $f(x)$  가 0인  $x$  값이 없다는 것입니다. 그러나  $x = 1.999999999999$ 에서는  $f(x)$ 에 대한 반대 부호가 생기도록 하는  $x$ 의 인접 값이 있습니다.

예:

방정식의 근을 구합니다.

$$\frac{x}{x^2 - 6} - 1 = 0$$

$x$ 가  $\sqrt{6}$ 에 이르면  $f(x)$ 가 매우 큰 양수 또는 음수가 됩니다.

방정식을 수식으로 입력합니다.

키:	디스플레이:	설명:
EQN		방정식 모드를 선택합니다.
RCL X $\div$ ( )		방정식을 입력합니다.
RCL X $y^x$ 2		
$\leftarrow$ 6 $\rightarrow$		
$\leftarrow$ 1 ENTER		
$\leftarrow$ SHOW	X $\div$ (X $^2$ -6)-1	체크섬 및 길이입니다.
	CK=7358	
	LN=11	
C		방정식 모드를 취소합니다.

이제 방정식을 풀어 근을 구합니다.

키:	디스플레이:	설명:
<b>2</b> <b>.</b> <b>3</b> <b>↵</b>		근의 초기 추정값입니다.
<b>STO</b> <b>X</b> <b>ENTER</b>	2.7_	
<b>2</b> <b>.</b> <b>7</b>		
<b>EQN</b>	$X \div (X^2 - 6) - 1$	방정식 모드를 선택하고 방정식을 표시합니다.
<b>↵</b> <b>SOLVE</b> <b>X</b>	NO ROOT FND	$f(x)$ 의 근이 없습니다.

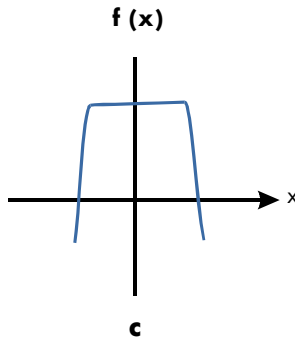
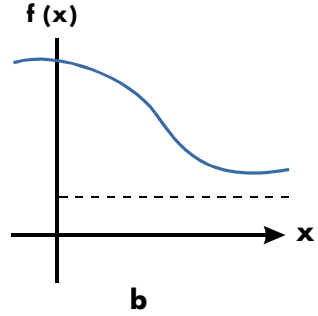
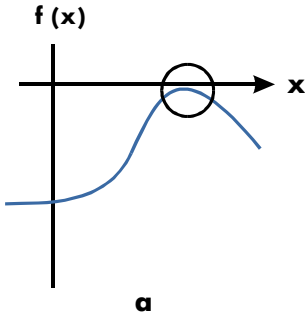
---

## SOLVE에서 근을 구할 수 없는 경우

SOLVE에서 근을 구하지 못하는 경우도 있습니다. 다음과 같은 경우 메시지 NO ROOT FND가 나타납니다.

- 지역 최소값 또는 최대값 근처에서 검색이 종료됩니다(아래의 그림 a 참조).
- $f(x)$ 가 기본적으로 넓은  $x$  범위에서 상수 상태를 유지하는 영역인 수평 점근선에서 SOLVE를 실행하기 때문에 검색이 중단됩니다(아래의 그림 b 참조).
- 함수의 "평평한" 지역 영역을 집중적으로 검색합니다(아래의 그림 c 참조).

이러한 경우 스택의 값이 SOLVE를 실행하기 전의 값과 동일하게 됩니다.



### Case Where No Root Is Found

#### 예: 상대 최소값

이 포물선 방정식의 근을 구합니다.

$$x^2 - 6x + 13 = 0.$$

최소값은  $x = 3$ 입니다.

방정식을 수식으로 입력합니다.

키:                      디스플레이:

$\boxed{\text{EQN}}$   
 $\boxed{\text{RCL}} \boxed{\text{X}} \boxed{y^x} \boxed{2}$   
 $\boxed{-} \boxed{6} \boxed{\text{X}} \boxed{\text{RCL}} \boxed{\text{X}}$   
 $\boxed{+} \boxed{1} \boxed{3} \boxed{\text{ENTER}}$        $x^2 - 6x + 13$

설명:  
 방정식 모드를 선택합니다.  
 방정식을 입력합니다.

**[↶] [SHOW]** CK=EC74  
LN=10

체크섬 및 길이입니다.

**[C]** 방정식 모드를 취소합니다.

이제 방정식을 풀어 근을 구합니다.

**키:**                      **디스플레이:**

**[0] [↵] [STO] [X]**  
**[ENTER] [1] [0]** 10\_  
**[EQN]** X^2-6×X+13

**설명:**  
근의 초기 추정값입니다.

**[↵] [SOLVE] [X]** NO ROOT FND

방정식 모드를 선택하고 방정식을 표시합니다.  
추정값 0 및 10을 사용하여 검색하지 못했습니다.

**예: 점근선**

방정식의 근을 구합니다.

$$10 - \frac{1}{X} = 0$$

방정식을 수식으로 입력합니다.

**키:**                      **디스플레이:**

**[EQN]**  
**[1] [0] [-] [1/x]**  
**[RCL] [X] [ENTER]** 10-INV(X)  
**[↶] [SHOW]** CK=6EAB  
LN=9

**설명:**  
방정식 모드를 선택합니다.  
방정식을 입력합니다.

**[C]**  
**[.] [0] [0] [5] [↵]**  
**[STO] [X] [ENTER] [5]** 5\_  
**[EQN]** 10-INV(X)

체크섬 및 길이입니다.  
방정식 모드를 취소합니다.  
근의 양수 추정값입니다.

**[↵] [SOLVE] [X]** X=  
0.1000  
0.1000  
0.000000000000

방정식 모드를 선택하고 방정식을 표시합니다.  
추정값 0.005 및 5를 사용하여 x의 값을 구합니다.  
이전 추정값이 동일합니다.  
f(x) = 0



**D-10 방정식 풀기 추가 정보**

음수 추정값을 사용할 때 발생하는 현상을 살펴 봅니다.

키:	디스플레이:	설명:
$\frac{+/-}{\square}$ <b>1</b> $\rightarrow$ <b>STO</b> $\square$ <b>X</b>	-1.0000	근의 음수 추정값입니다.
<b>ENTER</b>		
$\frac{+/-}{\square}$ <b>2</b> <b>EQN</b>	10-INV(X)	방정식 모드를 선택하고 방정식을 표시합니다.
$\rightarrow$ <b>SOLVE</b> $\square$ <b>X</b>	X= 0.1000	X의 값을 구하고 결과를 표시합니다.

**예:** 방정식의 근을 구합니다.

$$\sqrt{[x \div (x + 0.3)]} - 0.5 = 0$$

방정식을 수식으로 입력합니다.

키:	디스플레이:	설명:
<b>EQN</b>		방정식 모드를 선택합니다.
$\sqrt{x}$ <b>RCL</b> $\square$ <b>X</b> $\div$ $( )$		방정식을 입력합니다.
<b>RCL</b> $\square$ <b>X</b> <b>+</b> $\cdot$ <b>3</b>		
$>$ $>$ $-$ $\cdot$ <b>5</b>		
<b>ENTER</b>	SQRT(X÷(X+0.3)) $\rightarrow$	
$\leftarrow$ <b>SHOW</b>	CK=9F3B LN=19	체크섬 및 길이입니다.
<b>C</b>		방정식 모드를 취소합니다.

먼저 양수 근을 구합니다.

키:	디스플레이:	설명:
<b>0</b> $\rightarrow$ <b>STO</b> $\square$ <b>X</b>		근의 양수 추정값입니다.
<b>ENTER</b> <b>1</b> <b>0</b>	10_	
<b>EQN</b>	SQRT(X÷(X+0.3)) $\rightarrow$	방정식 모드를 선택하고 방정식의 왼쪽 끝을 표시합니다.
$\rightarrow$ <b>SOLVE</b> $\square$ <b>X</b>	X= 0.1000	추정값 0 및 10을 사용하여 근을 계산합니다.

이제 추정값 0 및 -10을 입력하여 음수 근을 구합니다. 0과 -0.3 사이의  $x$  값은 음수 분자가 아닌 양수 분모를 구하여 음수 제곱근이 나오도록 하기 때문에 이러한 값에 대해서는 함수가 정의되어 있지 않습니다.

키:	디스플레이:	설명:
<b>0</b> <b>→</b> <b>STO</b> <b>X</b>		
<b>ENTER</b> <b>+/-</b> <b>1</b> <b>0</b>	-10_	
<b>EQN</b>	$\text{SQRT}(X \div (X + 0.3))$	▶ 방정식 모드를 선택하고 방정식의 왼쪽 끝을 표시합니다.
<b>→</b> <b>SOLVE</b> <b>X</b>	NO ROOT FND	$f(x)$ 의 근이 없습니다.

### 예: "평평한" 지역 영역

함수의 근을 구합니다.

$$f(x) = x + 2 \text{ if } x < -1,$$

$$f(x) = 1 \text{ for } -1 \leq x \leq 1 \text{ (평평한 지역 영역),}$$

$$f(x) = -x + 2 \text{ if } x > 1.$$

RPN 모드에서 함수를 프로그램으로 입력합니다.

```
J001 LBL J
J002 1
J003 2
J004 RCL+ X
J005 x<y?
J006 RTN
J007 4
J008 -
J009 +/-
J010 x>y?
J011 R↓
J012 RTN
```

체크섬 및 길이: 9412 39



초기 추정값  $10^{-8}$  및  $-10^{-8}$ 을 사용하여 X의 값을 구합니다.

키:	디스플레이:	설명:
(RPN 모드)		
<b>1</b> <b>E</b> <b>8</b>		추정값을 입력합니다.
<b>+/−</b> <b>↶</b> <b>STO</b> <b>X</b> <b>1</b>	$-1E-8$	
<b>+/−</b> <b>E</b> <b>8</b> <b>+/−</b>		
<b>↵</b> <b>FN</b> <b>=</b> <b>J</b>	$-1.0000E-8$	프로그램 "J"를 함수로 선택합니다.
<b>↶</b> <b>SOLVE</b> <b>X</b>	X= $-2.0000$	X값을 구하고 결과를 표시합니다.

## 반올림 오차

계산기의 12자리 자릿수 제한으로 인해 반올림 오차가 발생할 수 있습니다. 이 경우 SOLVE 및 적분의 반복 해에 부정적인 영향을 줍니다. 예를 들어 다음과 같은 방정식에는

$$[(|x| + 1) + 10^{15}]^2 \cdot 10^{30} = 0$$

$f(x)$ 가 항상 0보다 크기 때문에 근이 없습니다. 그러나 초기 추정값으로 1 및 2가 제공되면 SOLVE에서 반올림 오차로 인해 결과값 1.0000이 반환됩니다.

반올림 오차로 인해 SOLVE에서 근을 구하지 못할 수도 있습니다. 다음 방정식의 근은

$$|x^2 - 7| = 0$$

$\sqrt{7}$ 입니다. 그러나  $\sqrt{7}$ 에 *정확하게* 일치하는 12자리 숫자가 없기 때문에 계산기에서 함수를 0에 일치시킬 수 없습니다. 또한 함수가 부호를 변경하지 않기 때문에 SOLVE에서 메시지 **NO ROOT FND**를 반환합니다.



## 적분 추가 정보

---

이 부록에서는 8장에 설명된 내용 이외의 적분 관련 정보를 제공합니다.

### 적분 계산 방법

적분 연산에 사용되는 알고리즘인  $\int_{FN} dx$ 는 적분 구간 내의 많은  $x$ (표본점) 값에서 함수 값의 가중 평균을 계산하여  $f(x)$  함수의 적분을 계산합니다. 이러한 표본추출 프로세스 결과의 정확성은 고려되는 표본점의 개수에 따라 달라집니다. 일반적으로 표본점이 많을수록 정확성이 높아집니다. 무한 개의 표본점으로  $f(x)$ 를 계산할 수 있는 경우 이 알고리즘은 계산된  $f(x)$  함수의 부정확성에 의한 제한을 무시하고 항상 정확한 결과를 제공합니다.

무한 개의 표본점으로 함수를 계산하는 작업은 영원히 끝나지 않습니다. 그러나 계산된 적분의 최대 정확도는 계산된 함수 값의 정확도에 의해 제한되기 때문에 이와 같은 작업은 수행되지 않습니다. 즉, 이 알고리즘은 정해진 개수의 표본점만 사용하여  $f(x)$ 의 내재된 불확실성에 비해 정확한 적분을 계산할 수 있습니다.

적분 알고리즘은 처음에 몇 개의 표본점만 고려하여 비교적 부정확한 근사값을 구합니다. 이러한 근사값이  $f(x)$ 에서 허용되는 수준만큼 정확하지 않는 경우 보다 많은 표본점을 사용하여 알고리즘이 반복됩니다. 결과 근사값이  $f(x)$ 의 내재된 불확실성에 비해 충분히 정확해질 때까지 매번 표본점의 개수를 약 두 배로 늘려서 이러한 반복이 계속됩니다.

8장에 설명된 것처럼 최종 근사값의 불확실성은 디스플레이 형식에서 파생된 숫자로 함수의 불확실성을 지정합니다. 각 반복이 끝날 때마다 이 알고리즘은 반복 중에 계산된 근사값을 이전의 두 반복에서 계산된 근사값과 비교합니다. 세 근사값 중 하나와 다른 두 근사값의 차이가 최종 근사값에서 허용되는 불확실성보다 작으면 계산이 끝나고 현재 근사값은 X 레지스터에, 해당 불확실성은 Y 레지스터에 유지됩니다.

세 개의 연속 근사값 각각의 오류, 즉 실제 적분과 근사값의 차이가 모두 근사값 간의 불일치보다 큰 경우는 거의 없습니다. 따라서  $f(x)$ 가 매우 빠르게 변화하지 않는 경우 최종 근사값의 오류는 해당 불확실성보다 작습니다. 최종 근사값의 오류를 알 수는 없지만 오류가 근사값의 표시된 불확실성을 초과할 가능성은 거의 없습니다. 즉, Y 레지스터의 불확실성 추정값은 근사값과 실제 적분의 차이에서 거의 확실한 "상한값"입니다.

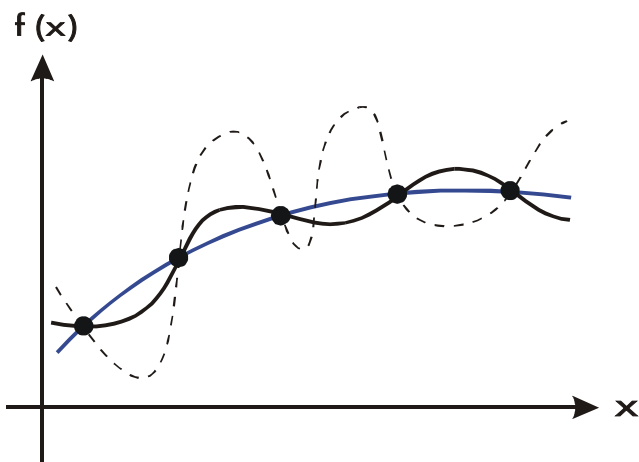
---

## 잘못된 결과가 나올 수 있는 경우

HP 35s의 적분 알고리즘은 최상의 알고리즘 중 하나이지만 특정 상황에서는 수치 적분에 대한 다른 모든 알고리즘과 마찬가지로 잘못된 답이 나올 수 있습니다.

*그러나 이러한 경우가 발생할 가능성은 거의 없습니다.* 이 알고리즘은 거의 모든 유연한 함수로 정확한 결과를 제공하도록 설계되었으며 매우 엉뚱한 동작을 보이는 함수에 대해서만 부정확한 결과를 얻을 위험이 있습니다. 이러한 함수는 실제 물리적 상황과 관련된 문제에서 거의 발생하지 않으며, 발생하는 경우 대체로 간단하게 함수를 인식하고 처리할 수 있습니다.

그러나 알고리즘이  $f(x)$ 에 대해 알고 있는 것은 표본점의 함수 값뿐이기 때문에 모든 표본점에서  $f(x)$ 에 일치하는 다른 함수와  $f(x)$ 를 구분할 수 없습니다. 다음 그림은 이러한 경우를 나타내며, 해당 그래프에 많은 표본점이 공통적으로 포함된 세 함수를 보여 줍니다(적분 구간의 일부임).



이 개수의 표본점을 사용하여 알고리즘은 표시된 모든 함수의 적분에 대해 동일한 근사값을 계산합니다. 파란색 실선과 검은색 실선으로 표시된 함수의 실제 적분은 대략 동일하므로  $f(x)$ 가 두 함수 중 하나인 경우 근사값이 꽤 정확합니다. 그러나 파선으로 표시된 함수의 실제 적분은 다른 두 함수와 크게 다르기 때문에  $f(x)$ 가 이 함수인 경우 현재 근사값이 비교적 부정확합니다.

알고리즘은 점점 더 많은 점의 함수를 표본추출하여 함수의 일반적인 동작을 알게 됩니다. 한 영역의 함수 변동이 적분 구간의 나머지 부분에서 표시된 동작과 다른 경우 몇 번 반복하면 대체로 알고리즘에서 변동을 탐지합니다. 이 경우 연속 반복을 통해 가장 빠르지만 *특정적인* 변동의 존재를 고려하는 근사값을 구할 때까지 표본점의 개수가 증가합니다.

예를 들어, 다음과 같은 방정식의 근사값을 고려해 보십시오.

$$\int_0^{\infty} x e^{-x} dx.$$

이 적분을 수치로 계산하고 있으므로 적분의 상한을 계산기에 입력할 수 있는 최대 숫자인  $10^{499}$ 으로 나타내야 한다고 생각할 수도 있습니다.

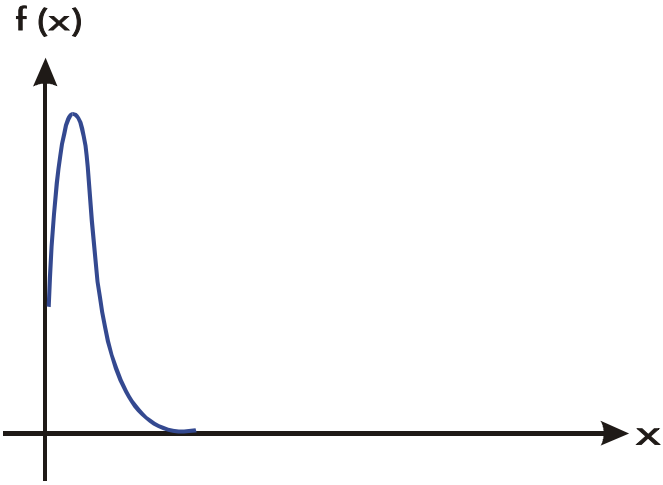
이 작업을 수행한 후 결과를 확인해 보십시오.  $f(x) = xe^{-x}$  함수를 입력합니다.

키:	디스플레이:	설명:
[EQN]		방정식 모드를 선택합니다.
[RCL] [X] [X] [e <sup>x</sup> ]	X×EXP( )	방정식을 입력합니다.
[+/-] [RCL] [X] [ENTER]	X×EXP(-X)	방정식의 끝입니다.
[↵] [SHOW]	CK=2FE6 LN=9	체크섬 및 길이입니다.
[C]		방정식 모드를 취소합니다.

디스플레이 형식을 SCI 3으로 설정하고, 적분의 하한과 상한을 0과 10499으로 지정한 다음 적분을 시작합니다.

키:	디스플레이:	설명:
✓ [↵] [DISPLAY] [2] (2SCI)		정확도 수준과 적분 한도를 지정합니다.
[3] [ENTER] [1] [E] [4]	1E499_	
[9] [9]		
[EQN]	X×EXP(-X)	방정식 모드를 선택하고 방정식을 표시합니다.
[↵] [∫] [X]	INTEGRATING ∫ = 0.000E0	적분의 근사값입니다.

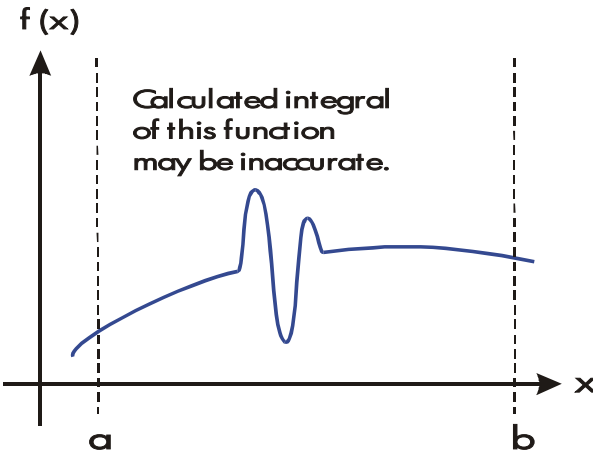
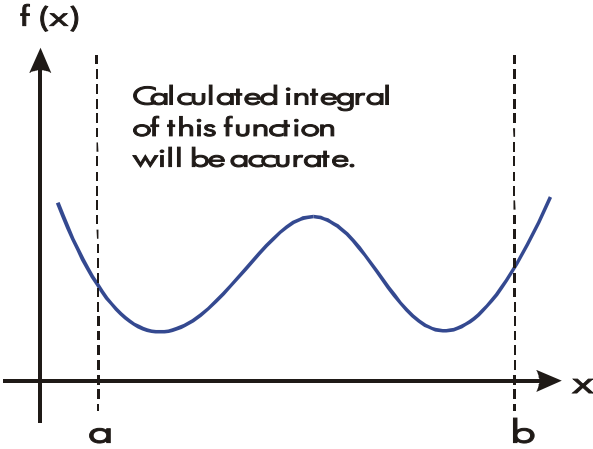
0에서 ∞까지  $f(x) = xe^{-x}$ 의 실제 적분은 정확하게 1이므로 계산기에서 반환된 결과는 분명히 잘못되었습니다. 그러나 0에서 10499까지 이 함수의 실제 적분이 1에 매우 가깝기 때문에 /∞가 10499으로 표현된 것은 문제가 되지 *않습니다*. 결과가 잘못된 이유는 적분 구간에 대한  $f(x)$  그래프에서 명확하게 나타납니다.



이 그래프는 원점에 매우 가까운 스파이크입니다. 스파이크를 발견한 표본점이 없기 때문에 알고리즘에서는  $f(x)$ 가 적분 구간 전체에서 동일하게 0과 같다고 가정했습니다. SCI 11 또는 ALL 형식으로 적분을 계산하여 표본점의 개수를 늘려도 이 특정 함수가 이 특정 구간에 대해 적분되는 경우 추가 표본점에서 스파이크를 발견하지 못합니다. 이러한 문제에 대한 더 나은 접근 방법은 다음 항목인 "계산 시간이 길어지는 경우"를 참조하십시오.

이러한 탈선(다른 곳에서는 함수 동작으로 나타나지 않는 변동)을 보이는 함수는 드물기 때문에 모르고 적분할 가능성이 거의 없습니다. 잘못된 결과를 초래할 수 있는 함수는 적분 구간에서 해당 함수와 낮은 자리 도함수가 얼마나 빠르게 변화하는지에 의해 간단한 용어로 식별될 수 있습니다. 기본적으로 함수 또는 해당 도함수의 변화가 빠르고 이처럼 빠르게 변화하는 도함수의 자리가 낮을수록 계산이 완료되는 데 오래 걸리고 결과 근사값의 신뢰도가 떨어집니다.

함수 또는 낮은 자리 도함수의 변화 속도는 적분 구간의 너비와 관련해서 확인해야 합니다. 지정된 개수의 표본점에서 3회의 변동이 있는  $f(x)$  함수는 이러한 변화가 구간의 일부분에만 국한된 경우보다 대부분의 적분 구간에 확산되어 있을 때 표본에 의해 보다 정확하게 표현될 수 있습니다. 이러한 두 경우는 다음 두 그림에 표시되어 있습니다. 변화 또는 변동을 함수의 진동 유형으로 고려할 경우 관련된 기준은 진동 주기와 적분 구간 너비의 비율입니다. 이 비율이 클수록 계산이 더 빨리 완료되고 결과 근사값의 신뢰도가 커집니다.





대부분의 경우 적분하려는 함수를 잘 알고 있으므로 적분 구간에 비해 함수에 빠른 변화가 있는지 여부를 파악할 수 있습니다. 함수를 잘 알지 못하며 이로 인해 문제가 발생한 것은 아닌지 의심되는 경우 해당 용도로 작성한 방정식이나 프로그램으로 함수를 계산하여 몇 개의 점을 신속하게 구성해 볼 수 있습니다.










적분의 근사값을 구한 후 어떤 이유로든 유효성이 의심되는 경우 간단한 절차를 통해 확인할 수 있습니다. 적분 구간을 둘 이상의 인접한 하위 구간으로 나누고 각 하위 구간에 대해 함수를 적분한 다음 결과 근사값을 더합니다. 이렇게 하면 함수가 새로운 표본점 집합에서 표본추출되어 이전에 숨겨진 스파이크를 표시할 가능성이 커집니다. 초기 근사값이 유효한 경우 하위 구간의 근사값 합계와 같을 것입니다.

## 계산 시간이 길어지는 경우

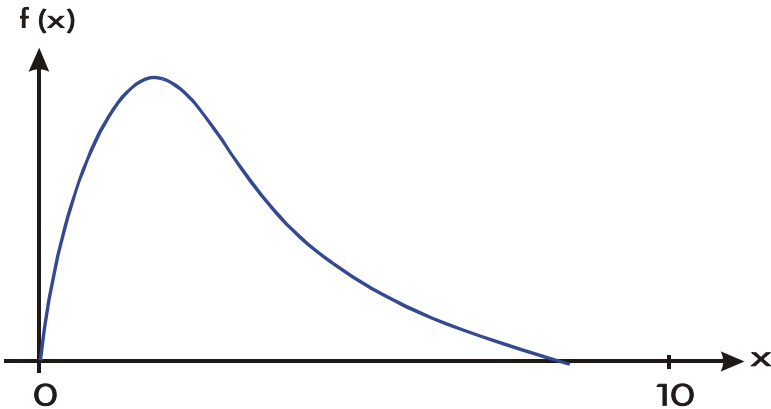
앞의 예에서 알고리즘은 함수의 스파이크를 탐지하지 못했기 때문에 잘못된 결과를 제공했습니다. 이 문제는 함수의 변화 속도가 적분 구간의 너비에 비해 너무 빨랐기 때문에 발생했습니다. 구간의 너비가 더 작으면 올바른 결과를 얻겠지만 구간이 여전히 넓으면 매우 오랜 시간이 걸립니다.

적분 구간이 초과 계산 시간이 필요할 만큼 넓지만 잘못 계산될 만큼 넓지는 않은 적분을 고려해 보십시오.  $f(x) = xe^{-x}$ 는  $x$ 가  $\infty$ 에 접근할 때 매우 빠르게 0에 접근하기 때문에 큰  $x$  값에서의 함수 적분에 대한 영향을 무시할 수 있습니다. 따라서 적분의 상한인  $\infty$ 를 10499만큼 크지 않은 숫자(예: 103)로 바꾸어 적분을 계산할 수 있습니다.

이 새로운 적분 한도를 사용하여 이전의 적분 문제를 다시 실행합니다.

키:	디스플레이:	설명:
   	1E3_	새로운 상한입니다.
 	$X \times \text{EXP}(-X)$	방정식 모드를 선택하고 방정식을 표시합니다.
  	INTEGRATING $\int =$ 1.0000E0	적분. 계산에 1분 또는 2분 정도 걸립니다.

이것은 올바른 결과이지만 시간이 많이 걸렸습니다. 이유를 이해하려면 이전 예에 표시된 그래프와 거의 동일한  $x=0$ 과  $x=10^3$  사이의 함수 그래프를  $x=0$ 과  $x=10$  사이의 함수 그래프와 비교합니다.



이 함수는 작은  $x$  값에서만 "의미가 있는 것"을 확인할 수 있습니다.  $x$  값이 크면 함수가 예측 가능한 방식으로 부드럽게 점점 감소하기 때문에 의미가 없습니다.

알고리즘은 연속 근사값 사이의 불일치가 충분히 작아질 때까지 표본점의 밀도를 더 높여서 함수를 표본추출합니다. 함수가 의미 있는 영역의 좁은 구간에서는 이 임계 밀도에 도달하는 시간이 더 짧습니다.

동일한 표본점 밀도를 얻기 위해 큰 구간에서 필요한 표본점의 총 개수가 작은 구간에서 필요한 개수보다 훨씬 더 많습니다. 따라서 큰 구간에서는 동일한 정확도의 근사값을 얻기 위해 더 많은 반복이 필요하므로 적분 계산에 더 오랜 시간이 걸립니다.

계산 시간은 함수가 의미가 있는 영역에서 특정 표본점 밀도에 얼마나 빨리 도달하는지에 따라 달라지므로 적분 구간에 포함된 대부분의 영역에서 함수가 의미가 없는 경우 모든 함수의 적분 계산이 연장됩니다. 이러한 적분을 계산해야 하는 경우 계산 시간이 상당히 단축되도록 문제를 수정할 수 있습니다. 두 가지 기술은 적분 구간 세분화와 변수 변형입니다. 두 방법을 사용하면 적분 구간에서 피적분 함수가 더 나은 동작을 보이도록 함수 또는 적분 한도를 변경할 수 있습니다.

## E-8 적분 추가 정보

## 메시지

---

계산기는 특정 상태나 키 입력에 메시지를 표시하여 응답합니다. 메시지를 확인 하도록 알려주는 ▲ 기호가 표시됩니다. 중요한 상태의 경우 사용자가 지울 때까지 메시지가 남아 있습니다. [C] 또는 [←]를 누르면 메시지가 지워지고 이전 디스플레이 내용이 표시됩니다. 아무 키나 눌러도 메시지가 지워지지만 키 기능은 실행되지 않습니다.

[FN ACTIVE

실행 중인 프로그램에서 적분 계산이 실행되는 동안 프로그램 레이블(FN=레이블)을 선택하려고 했습니다.

[<[FN)

실행 중인 프로그램에서 다른 적분 계산이 실행되는 동안 프로그램([FN d 변수)을 적분하려고 했습니다.

[<[SOLVE)

실행 중인 프로그램에서 적분 계산이 실행되는 동안 프로그램을 해결하려고 했습니다.

ALL VARS=0

변수 카탈로그([←] [MEM] [1] (1VAR))에서 저장된 값이 없음을 나타냅니다.

BAD GUESS

방정식에서 변수 값을 구할 때 잘못된 추정값(예: 복소수 또는 벡터)을 설정했습니다.

CALCULATING

계산기에서 시간이 오래 걸릴 수 있는 함수를 실행하고 있습니다.

CLR ALL? Y N

메모리에 있는 모든 내용을 지울지를 확인합니다.

CLR EQN? Y N

편집 중인 방정식을 지울지를 확인합니다. 방정식 입력 모드에서만 발생합니다.

CLR PGMS? Y N

메모리에 있는 모든 프로그램을 지울지를 확인합니다. 프로그램 입력 모드에서만 발생합니다.

DIVIDE BY 0	0으로 나누려고 했습니다. Y 레지스터에 0이 포함되어 있는 경우 <b>[←] [%CHG]</b> 를 포함합니다.
DUPLICAT.LBL	다른 프로그램 루틴에 대해 이미 존재하는 프로그램 레이블을 입력하려고 했습니다.
EQN LIST TOP	방정식 메모리의 "맨 위"를 나타냅니다. 메모리 체계가 순환되므로 EQN LIST TOP도 방정식 메모리에 있는 마지막 방정식 뒤의 "방정식"입니다.
INTEGRATING	계산기에서 방정식 또는 프로그램의 적분을 계산하고 있습니다. <i>시간이 오래 걸릴 수 있습니다.</i>
INTERRUPTED	실행 중인 CALCULATE, SOLVE 또는 ∫ FN 연산이 ALG, RPN, EQN 또는 PGM 모드에서 <b>[C]</b> 또는 <b>[R/S]</b> 를 눌러 중단되었습니다.
INVALID DATA	데이터 오류: <ul style="list-style-type: none"> <li>■ 오류 데이터를 저장하거나 계산하려고 했습니다.</li> <li>■ <math>r &gt; n</math>, 정수가 아닌 <math>r</math>이나 <math>n</math> 또는 <math>n \geq 1016</math>을 사용하여 조합 또는 순열을 계산하려고 했습니다.</li> <li>■ 통계 데이터에 복소수나 벡터를 저장하려고 했습니다.</li> <li>■ 포함된 자릿수가 허용되는 최대 밀수 <math>n</math> 자릿수보다 큰 밀수 <math>n</math>을 저장하려고 했습니다.</li> <li>■ <b>[x↔y]</b> 연산을 사용하여 통계 레지스터에 잘못된 데이터를 저장하려고 했습니다.</li> <li>■ 복소수나 벡터를 비교하려고 했습니다.</li> <li>■ 삼각 함수 또는 쌍곡선 함수를 잘못된 인수와 함께 사용하려고 했습니다. <ul style="list-style-type: none"> <li>■ <b>[TAN]</b> (<math>x</math>는 <math>90^\circ</math>의 홀수 배수)</li> <li>■ <b>[↔] [ACOS]</b> 또는 <b>[↔] [ASIN]</b> (<math>x &lt; -1</math> or <math>x &gt; 1</math>)</li> <li>■ <b>[←] [HYP] [↔] [ATAN]</b> (<math>x \leq -1</math> 또는 <math>x \geq 1</math>)</li> <li>■ <b>[←] [HYP] [↔] [ACOS]</b> (<math>x &lt; 1</math>)</li> </ul> </li> </ul>

INVALID VAR	방정식을 풀 때 잘못된 변수 이름을 입력하려고 했습니다.
INVALID *!	x를 음의 정수로 사용하여 계승 또는 감마 연산을 시도했습니다.
INVALID v*	지수 오류: <ul style="list-style-type: none"> <li>■ 0의 0 제곱 또는 음수 거듭제곱을 구하려고 했습니다.</li> <li>■ 음수의 비정수 거듭제곱을 구하려고 했습니다.</li> <li>■ 실수부가 음수인 숫자에 대해 복소수 (0 + i0)을 구하려고 했습니다.</li> </ul>
INVALID (I)	잘못된 간접 값으로 연산을 시도했습니다((I)가 정의되지 않음).
INVALID (J)	잘못된 간접 값으로 연산을 시도했습니다((J)가 정의되지 않음).
LOG(0)	0 또는 (0 + i0)의 로그를 구하려고 했습니다.
LOG<NEG>	음수의 로그를 구하려고 했습니다.
MEMORY CLEAR	사용자 메모리가 모두 지워졌습니다(페이지 참조).
MEMORY FULL	계산기의 메모리가 부족하여 연산을 수행할 수 없습니다(부록 B 참조).
NO	테스트 명령에 의해 확인된 조건이 참이 아닙니다. 키보드에서 실행된 경우에만 발생합니다.
NONEXISTENT	[GTO], [XEQ] 또는 FN을 사용하여 존재하지 않는 프로그램 레이블(또는 줄 번호)을 참조하려고 했습니다. NONEXISTENT 오류는 다음을 의미할 수 있습니다. <ul style="list-style-type: none"> <li>■ 존재하지 않는 프로그램 레이블을 키보드에서 명시적으로 호출했습니다.</li> <li>■ 호출한 프로그램에서 존재하지 않는 다른 레이블을 참조했습니다.</li> </ul> 적분 결과가 존재하지 않습니다.

NO LABELS	프로그램 카탈로그( <b>☐</b> <b>MEM</b> <b>2</b> (2PGM) )에서 저장된 프로그램 레이블이 없음을 나타냅니다.
NO SOLUTION	이 선형 방정식 체계에 대한 해를 찾을 수 없습니다.
MULT SOLUTION	이 선형 방정식 체계에 대한 여러 개의 해를 찾았습니다.
NO ROOT FND	SOLVE(EQN 및 PGM 모드 포함)에서 현재 초기 추측을 사용하여 방정식의 근을 구할 수 없습니다(페이지 참조). 이러한 조건에는 잘못된 추측, 풀 수 없음, 관심 점, 왼쪽과 오른쪽이 일치하지 않음 등이 포함됩니다. 프로그램에서 실행된 SOLVE 연산은 이 오류를 생성하지 않습니다. 대신 동일한 상태에서 다음 프로그램 줄(SOLVE 변수 명령 다음 줄)을 건너뛸니다.
OVERFLOW	경고(잠시 표시됨). 결과 크기가 너무 커서 계산기에서 처리할 수 없습니다. 계산기는 현재 디스플레이 형식으로 ±9.999999999999E499를 반환합니다. 페이지의 "숫자 범위와 오버플로"를 참조하십시오. 이 상태는 플래그 6을 설정합니다. 플래그 5가 설정되면 오버플로에 실행 중인 프로그램을 중지하고 사용자가 아무 키나 누를 때까지 메시지가 디스플레이에 계속 표시되게 하는 추가 효과가 있습니다.
PRGM TOP	프로그램 메모리의 "맨 위"를 나타냅니다. 메모리 체계가 순환되므로 PRGM TOP도 프로그램 메모리에 있는 마지막 줄 뒤의 "줄"입니다.
RUNNING <b>B</b>	계산기에서 SOLVE 또는 fFN 루틴이 아닌 방정식 또는 프로그램을 실행하고 있습니다.
SELECT FN	선택한 프로그램 레이블 없이 SOLVE 변수 또는 fFN d 변수를 실행하려고 했습니다. 이 메시지는 MEMORY CLEAR 메시지 후에 처음 SOLVE 또는 fFN을 사용할 때 발생하거나 현재 레이블이 더 이상 존재하지 않는 경우에 발생할 수 있습니다.
SOLVE ACTIVE	실행 중인 프로그램에서 SOLVE 연산이 실행되는 동안 프로그램 레이블(FN= 레이블)을 선택하려고 했습니다.

SOLVE(SOLVE)	실행 중인 프로그램에서 SOLVE 연산이 실행되는 동안 프로그램을 해결하려고 했습니다.
SOLVE(∫ FN)	실행 중인 프로그램에서 SOLVE 연산이 실행되는 동안 프로그램을 적분하려고 했습니다.
SOLVING	계산기에서 근을 구하기 위해 방정식 또는 프로그램을 풀고 있습니다. 시간이 오래 걸릴 수 있습니다.
SQRT(NEG)	음수의 제곱근을 계산하려고 했습니다.
STAT ERROR	통계 오류: <ul style="list-style-type: none"> <li>■ <math>n=0</math>을 사용하여 통계 계산을 수행하려고 했습니다.</li> <li>■ <math>n=1</math>을 사용하여 <math>s_x, s_y, \hat{x}, \hat{y}, m, r</math> 또는 <math>b</math>를 계산하려고 했습니다.</li> <li>■ <math>x</math>데이터만 사용하여 <math>r, \hat{x}</math> 또는 <math>\bar{x}w</math>를 계산하려고 했습니다(모든 <math>y</math>값은 0과 같음).</li> <li>■ 모든 <math>x</math>값을 동일하게 지정하여 <math>\hat{x}, \hat{y}, r, m</math> 또는 <math>b</math>를 계산하려고 했습니다.</li> </ul>
SYNTAX ERROR	수식, 방정식, <b>SOLVE</b> 또는 <b>∫</b> 를 계산하는 동안 구문 오류가 탐지되었습니다. <b>←</b> 또는 <b>C</b> 를 누르면 오류 메시지가 지워지고 오류를 수정할 수 있습니다.
TOO BIG	숫자의 크기가 너무 커서 HEX, OCT 또는 BIN 밑으로 변환할 수 없습니다. 숫자가 $-34,359,738,368 \leq n \leq 34,359,738,367$ 범위 내에 있어야 합니다.
XEQ OVERFLOW	실행 중인 프로그램에서 XEQ 레이어/블을 21번째 중첩하려고 했습니다. 최대 20개의 서브루틴을 중첩할 수 있습니다. SOLVE와 ∫ FN도 각각 수준을 사용하므로 이 오류를 생성할 수 있습니다.
YES	테스트 명령에 의해 확인된 조건이 참입니다. 키보드에서 실행된 경우에만 발생합니다.

## 자가 테스트 메시지:

35S-OK

자가 테스트와 키보드 테스트에 통과했습니다.

35S-FAIL *n*

자가 테스트 또는 키보드 테스트가 실패했으며 계산기를 수리해야 합니다.



© 2007 HP DEV CO. L. P.


자가 테스트를 성공적으로 완료한 후 저작권 메시지가 표시됩니다.



## 연산 색인








이 절에서는 모든 함수와 연산, 그리고 해당되는 경우 공식에 대한 빠른 참조를 제공합니다. 목록은 함수 이름을 기준으로 알파벳순으로 표시됩니다. 이 이름은 프로그램 줄에서 사용된 이름입니다. 예를 들어, FIX  $n$ 이라는 함수는

 **DISPLAY**  (FIX)  $n$ 으로 실행됩니다.

프로그램 가능하지 않은 함수의 이름은 키 상자에 있습니다. 예를 들면, 와 같습니다.

글자가 아닌 문자와 그리스어 문자는 모든 글자 앞에 알파벳순으로 표시됩니다. 앞에 화살표가 있는 함수 이름(예: →DEG)은 화살표가 없는 것처럼 알파벳순으로 표시됩니다.

\*로 표시된 마지막 열은 테이블 끝에 있는 참고를 나타냅니다.

이름	키 및 설명	페이지	*
+/-	 숫자의 부호를 바꿉니다.	1-15	1
+	 덧셈. $y + x$ 값을 반환합니다.	1-19	1
-	 뺄셈. $y - x$ 값을 반환합니다.	1-19	1
×	 곱셈. $y \times x$ 값을 반환합니다.	1-19	1
÷	 나눗셈. $y \div x$ 값을 반환합니다.	1-19	1
^	 거듭제곱. 지수를 나타냅니다.	6-16	1
	입력된 마지막 숫자를 삭제하고, $x$ 를 지우고, 메뉴를 지우고, 방정식에 입력된 마지막 함수를 지우고, 방정식을 삭제하고, 프로그램 단계를 삭제합니다.	1-4 1-8 6-3 13-7	
























이름	키 및 설명	페이지	*
	카탈로그의 이전 항목을 표시하고, 방정식 목록에서 이전 방정식으로 이동하고, 프로그램 포인터를 이전 단계로 이동합니다.	1-28 6-3 13-11 13-20	
	카탈로그의 다음 항목을 표시하고, 방정식 목록에서 다음 방정식으로 이동하고, 프로그램 입력 중 프로그램 포인터를 다음 줄로 이동하고, 프로그램 입력 중이 아닌 경우 현재 프로그램 줄을 실행합니다.	1-28 6-3 13-11 13-20	
또는	커서를 이동하고 내용은 삭제하지 않습니다.	1-14	
또는	디스플레이를 스크롤하여 왼쪽 및 오른쪽의 나머지 숫자를 표시하고, 방정식 또는 2진수의 나머지 값을 표시하고, CONST 및 SUMS 메뉴의 다음 메뉴 페이지로 이동합니다.	1-11 6-4 11-8	
	방정식의 맨 윗줄이나 프로그램 모드에서 마지막 레이블의 첫 번째 줄로 이동합니다.	6-3	
	방정식의 마지막 줄이나 프로그램 모드에서 다음 레이블의 첫 번째 줄로 이동합니다.	6-3	
,	함수에서 둘 또는 세 인수를 구분합니다.	6-5	1
1/x	역수	1-18	1
10 <sup>x</sup>	상용 지수 10의 x 제곱을 반환합니다.	4-2	1
%	백분율 (y × x) ÷ 100을 반환합니다.	4-6	1

이름	키 및 설명	페이지	*
%CHG	 <b>%CHG</b> 백분을 변경 ( $x - y$ )(100 ÷ $y$ ) 값을 반환합니다.	4-6	1
$\pi$	 <b><math>\pi</math></b> 근사값 3.14159265359(12자리수)를 반환합니다.	4-3	1
$\Sigma+$	 <b><math>\Sigma+</math></b> ( $y, x$ )를 통계 레지스터에 누적합니다.	12-2	
$\Sigma-$	 <b><math>\Sigma-</math></b> ( $y, x$ )를 통계 레지스터에서 제거합니다.	12-2	
$\Sigma x$	 <b>SUMS</b>  ( <b><math>\Sigma x</math></b> ) $x$ 값의 합계를 반환합니다.	12-11	1
$\Sigma x^2$	 <b>SUMS</b>    ( <b><math>\Sigma x^2</math></b> ) $x$ 값의 제곱 합계를 반환합니다.	12-11	1
$\Sigma xy$	 <b>SUMS</b>       ( <b><math>\Sigma xy</math></b> ) $x$ 값과 $y$ 값의 곱 합계를 반환합니다.	12-11	1
$\Sigma y$	 <b>SUMS</b>   ( <b><math>\Sigma y</math></b> ) $y$ 값의 합계를 반환합니다.	12-11	1
$\Sigma y^2$	 <b>SUMS</b>     ( <b><math>\Sigma y^2</math></b> ) $y$ 값의 제곱 합계를 반환합니다.	12-11	1
$\sigma x$	 <b>S.<math>\sigma</math></b>   ( <b><math>\sigma x</math></b> ) $x$ 값의 모집단 표준 편차를 반환합니다.	12-7	1
$\sigma y$	 <b>S.<math>\sigma</math></b>    ( <b><math>\sigma y</math></b> ) $y$ 값의 모집단 표준 편차를 반환합니다.	12-7	1
	$\sqrt{\sum (x_i - \bar{x})^2 \div n}$		
	$\sqrt{\sum (y_i - \bar{y})^2 \div n}$		

이름	키 및 설명	페이지	*
$\int$ FN d 변수	( $\int$ FN d $\int$ ) 변수 Y 레지스터의 적분 변수 하한과 X 레지스터의 적분 변수 상한을 사용하여 FN=에 의해 선택된 프로그램이나 표시된 방정식을 적분합니다.	8-2 15-7	
( )	괄호. 추가 계산을 위해 괄호를 그대로 두려면 를 누릅니다.	6-6	1
[ ]	: 벡터 연산을 수행하기 위한 벡터 기호입니다.	10-1	1
$\theta$	$\theta$ : 복소수 연산을 수행하기 위한 복소수 기호입니다.	9-1	1
A through Z	변수 명명된 변수의 값입니다.	6-4	1
ABS	절대값 $ x $ 를 반환합니다.	4-17	1
ACOS	아크코사인 $\cos^{-1}x$ 를 반환합니다.	4-4	1
ACOSH	쌍곡선 아크코사인 $\cosh^{-1} x$ 를 반환합니다.	4-6	1
(4RLG)	대수 모드를 활성화합니다.	1-9	
ALOG	상용 지수 10의 지정된 거듭제곱(역대수)을 반환합니다.	6-16	1
ALL	(4RLL) 의미있는 모든 숫자를 표시합니다. 모든 숫자를 표시하기 위해 오른쪽으로 ( ) 스크롤해야 할 수도 있습니다.	1-23	

이름	키 및 설명	페이지	*
AND	LOGIC  (1AND) 논리 연산자	11-4	1
ARG	ARG 복소수를 "θ" 인수로 대체합니다.	4-17	1
ASIN	ASIN 아크사인 $\sin^{-1} x$ 를 반환합니다.	4-4	1
ASINH	HYP  ASIN 쌍곡선 아크사인 $\sinh^{-1} x$ 를 반환합니다.	4-6	1
ATAN	ATAN 아크탄젠트 $\tan^{-1} x$ 를 반환합니다.	4-4	1
ATANH	HYP  ATAN 쌍곡선 아크탄젠트 $\tanh^{-1} x$ 를 반환합니다.	4-6	1
$b$	L.R.     ( $b$ ) 회귀선의 $y$ 절편을 반환합니다. $\bar{y} - m\bar{x}$	12-11	1
$b$	BASE  (8b) 2진수를 나타냅니다.	11-2	1
BASE	진법 변환 메뉴를 표시합니다.	11-1	
BIN	BASE  (4BIN) 2진수(밑 2) 모드를 선택합니다.	11-1	
	계산기를 켜고, $x$ 를 지우고, 메시지 및 프롬프트를 지우고, 메뉴를 취소하고, 카탈로그를 취소하고, 방정식 입력을 취소하고, 프로그램 입력을 취소하고, 방정식 실행을 중지하고, 실행 중인 프로그램을 중지합니다.	1-1 1-4 1-8 1-29 6-3 13-7 13-19	

이름	키 및 설명	페이지	*
/c	분모 표시된 분수의 분모 한도를 x로 설정합니다. x=1이면 현재 /c 값을 표시합니다.	5-4	
→°C	°F를 °C로 변환합니다.	4-14	1
CF n	<b>FLAGS</b> <b>2</b> (2CF) n 플래그 n(n=0-11)을 지웁니다.	14-12	
<b>CLEAR</b>	숫자나 메모리의 일부를 지우고, 표시된 변수 또는 프로그램을 MEM 카탈로그에서 지우고, 표시된 방정식을 지우는 메뉴를 표시합니다.	1-5 1-28	
<b>CLEAR</b> <b>3</b> (3ALL)	저장된 모든 데이터, 방정식 및 프로그램을 지웁니다.	1-29	
<b>CLEAR</b> <b>3</b> (3PGM)	모든 프로그램을 지웁니다(프로그램 모드의 계산기).	13-23	
<b>CLEAR</b> <b>3</b> (3EQN)	표시된 방정식을 지웁니다(방정식 모드의 계산기).	13-7	
CLS	<b>CLEAR</b> <b>4</b> (4Σ) 통계 레지스터를 지웁니다.	12-1	
CLVARS	<b>CLEAR</b> <b>2</b> (2VARS) 모든 변수를 0으로 지웁니다.	3-6	
CLx	<b>CLEAR</b> <b>1</b> (1X) x(X 레지스터)를 0으로 지웁니다.	2-3 2-7 13-7	
CLVARx	<b>CLEAR</b> <b>6</b> (6CLVARx) 주소가 x 주소보다 큰 간접 변수를 0으로 지웁니다.	1-4	
CLSTK	<b>CLEAR</b> <b>5</b> (5STK) 모든 스택 수준을 0으로 지웁니다.	2-7	



















이름	키 및 설명	페이지	*
→CM	  <b>cm</b> 인치를 센티미터로 변환합니다.	4-14	1
nCr	  <b>nCr</b> 동시에 $n$ 개의 항목 중 $r$ 개를 선택하는 조합 $n! \div (r!(n-r)!)$ 를 반환합니다.	4-15	1
COS	 <b>COS</b> <i>코사인</i> $\cos x$ 를 반환합니다.	4-3	1
COSH	  <b>HYP</b>  <b>COS</b> 쌍곡선 <i>코사인</i> . $\cosh x$ 를 반환합니다.	4-6	1
 <b>CONST</b>	41개의 물리 상수에 액세스합니다.	4-8	
d	  <b>BASE</b>  <b>5</b> (5d) 10진수를 나타냅니다.	11-1	1
DEC	  <b>BASE</b>  <b>1</b> (1DEC) 10진수 모드를 선택합니다.	11-1	
DEG	  <b>MODE</b>  <b>1</b> (1DEG) 도 각도 모드를 선택합니다.	4-4	
→DEG	  <b>DEG</b> <i>라디안을 도로 변환</i> $(360/2\pi) x$ 를 반환합니다.	4-13	1
 <b>DISPLAY</b>	디스플레이 형식, 기수( $\cdot$ 또는 $,$ ), 천 단위 구분 기호 및 복소수 디스플레이 형식을 설정하는 메뉴를 표시합니다.	1-21	
DSE 변수	  <b>DSE</b> <i>변수</i> <i>감소, 작거나 같으면 건너뛴</i> . 변수에 저장된 제어 숫자 $cccccc.fff$ 에 대해 $i$ (증가 값)를 $cccccc$ (카운터 값)에서 빼고, 결과 $\leq fff$ (최종 값)이면 다음 프로그램 줄을 건너뛵니다.	14-18	

이름	키 및 설명	페이지	*
<b>E</b>	지수 입력을 시작하고 입력되는 숫자에 "E"를 추가합니다. 뒤에 10의 거듭제곱이 표시됨을 나타냅니다.	1-15	1
ENG n	<b>←</b> <b>DISPLAY</b> <b>3</b> ( <b>3ENG</b> ) <i>n</i> 첫 번째 숫자 뒤에 <i>n</i> 자릿수가 표시되는 과학 표시를 선택합니다 ( <i>n</i> = 0-11).	1-22	
<b>←ENG</b> / <b>ENG→</b>	표시 중인 숫자의 지수 표시가 3의 배수로 변경되도록 합니다.	1-22	
<b>ENTER</b>	연속해서 입력된 두 숫자를 구분하고, 방정식 입력을 완료하고, 표시된 방정식을 계산하고, 해당되는 경우 결과를 저장합니다.	1-19 6-4 6-11	
ENTER	<b>ENTER</b> <i>x</i> 를 Y 레지스터에 복사하고, <i>y</i> 를 Z 레지스터로 올리고, <i>z</i> 를 T 레지스터로 올리고, <i>t</i> 를 버립니다.	2-6	
<b>EQN</b>	방정식 입력 모드를 활성화하거나 취소합니다(토글).	6-3 13-7	
e <sup>x</sup>	<b>→</b> <b>e<sup>x</sup></b> 자연 지수 <i>e</i> 의 <i>x</i> 제곱을 반환합니다.	4-1	1
EXP	<b>→</b> <b>e<sup>x</sup></b> 자연 지수 <i>e</i> 의 지정된 거듭제곱을 반환합니다.	6-16	1
→°F	<b>←</b> <b>←°F</b> °C를 °F로 변환합니다.	4-14	1
<b>→</b> <b>FDISP</b>	분수 디스플레이 모드를 켜고 끕니다.	5-1	
FIX n	<b>DISPLAY</b> <b>1</b> ( <b>1FIX</b> ) <i>n</i> 소수 <i>n</i> 자릿수의 고정 디스플레이를 선택합니다. 0 ≤ <i>n</i> ≤ 11	1-21	



이름	키 및 설명	페이지	*
<b>FLAGS</b>	플래그를 설정하고 지우고 테스트하는 메뉴를 표시합니다.	14-12	
FN = 레이블	<b>FN=</b> 레이블 레이블이 지정된 프로그램을 SOLVE 및 J FN에서 사용되는 현재 함수로 선택합니다.	15-1 15-7	
FP	<b>INTG</b> <b>5</b> (5FP) $x$ 의 소수부	4-17	1
FS? $n$	<b>FLAGS</b> <b>3</b> (3FS?) $nn$ 플래그 $n$ ( $n = 0-11$ )을 설정하면 다음 프로그램 줄을 실행하고, 플래그 $n$ 을 지우면 다음 프로그램 줄을 건너뛸니다.	14-12	
→GAL	리터를 괄론으로 변환합니다.	4-14	1
GRAD	<b>MODE</b> <b>3</b> (3GRD) 그래드 각도 모드를 설정합니다.	4-4	
레이블 $nnn$	프로그램 포인터를 프로그램 레이블의 줄 $nnn$ 으로 설정합니다.	13-21	
	프로그램 포인터를 PRGM TOP으로 설정합니다.	13-21	
h	<b>BASE</b> <b>6</b> (6h) 16진수를 나타냅니다.	11-1	1
HEX	<b>BASE</b> <b>2</b> (2HEX) 16진수(밑 16) 모드를 선택합니다.	11-1	
<b>HYP</b>	쌍곡선 함수에 대해 HYP_ prefix를 표시합니다.	4-6	
→HMS	 시간을 시간/분/초로 변환 $x$ 를 소수에서 시간-분-초 형식으로 변환합니다.	4-13	1

이름	키 및 설명	페이지	*
HMS→	<b>HMS→</b> 시간/분/초를 시간으로 변환. x를 시간-분-초 형식에서 소수로 변환합니다.	4-13	1
(I)/(J)	복소수 입력에 사용됩니다.	9-2	1
→IN	<b>(I)</b> / <b>(J)</b> , <b>(I)</b> / <b>(J)</b> . 글자가 I/J 변수에 저장된 숫자 값에 해당하는 변수의 값입니다.	6-4 14-21	1
→IN	센티미터를 인치로 변환합니다.	4-14	1
IDIV	<b>2</b> ( <b>2INT+</b> ) 두 정수를 사용한 나눗셈 연산의 몫을 구합니다.	6-16	1
INT÷	<b>2</b> ( <b>2INT+</b> ) 두 정수를 사용한 나눗셈 연산의 몫을 구합니다.	4-2	1
INTG	<b>4</b> ( <b>4INTG</b> ) 지정한 숫자보다 작거나 같은 가장 큰 정수를 구합니다.	4-18	1
INPUT 변수	변수 변수를 X 레지스터로 다시 호출하고, 변수의 이름과 값을 표시하고, 프로그램 실행을 중지합니다.  (프로그램 실행 다시 시작) 또는  (현재 프로그램 줄 실행)를 누르면 입력이 변수에 저장됩니다. 프로그램에서만 사용됩니다.	13-13	
INV	인수의 역수	6-16	1
IP	<b>6</b> ( <b>6IP</b> ) x의 정수부	4-17	1

이름	키 및 설명	페이지	*
ISG 변수	 <b>ISG</b> 변수 증가, 크면 건너뛰 변수에 저장된 제어 숫자 cccccc.fff <i>i</i> 에 대해 <i>i</i> (증가 값)를 cccccc(카운터 값)에 더하고, 결과 > fff(최종 값)이면 다음 프로그램 줄을 건너뜹니다.	14-18	
→KG	  파운드를 킬로그램으로 변환합니다.	4-14	1
→KM	  마일을 킬로미터로 변환 합니다.	4-14	1
→L	  갤론을 리터로 변환합니다.	4-14	1
LAST <i>x</i>	 <b>LAST<i>x</i></b> LAST X 레지스터에 저장된 숫자를 반환합니다.	2-8	
→LB	  킬로그램을 파운드로 변환합니다.	4-14	1
LBL 레이블	 <b>LBL</b> 레이블 XEQ, GTO 또는 FN= 연산에서 참조 하기 위해 프로그램에 단일 글자의 레이블을 지정합니다. 프로그램에서 만 사용됩니다.	13-3	
LN	 <b>LN</b> 자연 로그 $\log_e x$ 를 반환합니다.	4-1	1
LOG	 <b>LOG</b> 상용 로그 $\log_{10} x$ 를 반환합니다.	4-1	1
 <b>L.R.</b>	선형 회귀 메뉴를 표시합니다.	12-4	
m	 <b>L.R.</b>    (m) 회귀선의 기울기를 반환합니다. $[\Sigma(x_i - \bar{x})(y_i - \bar{y})] \div \Sigma(x_i - \bar{x})^2$	12-7	1

이름	키 및 설명	페이지	*
→MILE	MILE 키로 킬로미터를 마일로 변환합니다.	4-14	1
MEM	사용 가능한 메모리 크기와 카탈로그 메뉴를 표시합니다.	1-28	
MEM 2 (2PGM)	프로그램 카탈로그를 시작합니다.	13-22	
MEM 1 (1VAR)	변수 카탈로그를 시작합니다.	3-4	
MODE	ALG 또는 RPN 모드나 각도 모드를 설정하는 메뉴를 표시합니다.	1-7 4-4	
n	SUMS (n) 데이터 요소 세트의 개수를 반환합니다.	12-11	1
NAND	LOGIC 5 (5NAND) 논리 연산자	11-4	1
NOR	LOGIC 6 (6XOR) 논리 연산자	11-4	1
NOT	LOGIC 4 (4NOT) 논리 연산자	11-4	1
o	BASE 7 (7o) 8진수를 나타냅니다.	11-2	1
OCT	BASE 3 (3OCT) 8진수(밑 8) 모드를 선택합니다.	11-1	
OR	LOGIC 3 (3OR) 논리 연산자	11-4	1
OFF	계산기를 끕니다.	1-1	
nPr	nPr 동시에 n개의 항목 중 r개를 선택하는 순열. $n \neq (n - r)!$ 를 반환합니다.	4-15	1

이름	키 및 설명	페이지	*
PRGM	프로그램 입력 모드를 활성화하거나 취소합니다(토글).	13-6	
PSE	PSE 일시 중지 $x_i$ 변수 또는 방정식을 표시하기 위 해 프로그램 실행을 잠시 중지한 후 다시 시작합니다. 프로그램에서만 사용됩니다.	13-18 13-19	
r	LR.   (r) $x$ 값과 $y$ 값 사이의 상관 계수를 반환합니다. $\frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \times (y_i - \bar{y})^2}}$	12-7	1
$\rho\theta\alpha$	DISPLAY   (10rθα) 복소수 표시를 변경합니다.	1-25	
RAD	MODE  (1RAD) 라디안 각도 모드를 선택합니다.	4-4	
→RAD	→RAD 도를 라디안으로 변환 ( $2\pi/360$ ) $x$ 를 반환합니다.	4-13	1
RADIX ,	DISPLAY  (6,) 십표를 기수 표시(소수점)로 선택합 니다.	1-23	
RADIX .	DISPLAY  (5.) 마침표를 기수 표시(소수점)로 선택 합니다.	1-23	
RANDOM	RAND RANDOM 함수를 실행 합니다. 0에서 1 사이의 난수를 반 환합니다.	4-15	1
RCL 변수	변수 다시 호출 변수를 X 레지스터에 복사합니다.	3-7	






























이름	키 및 설명	페이지	*
RCL+ 변수	<b>RCL</b> <b>+</b> 변수 $x +$ 변수를 반환합니다.	3-7	
RCL- 변수	<b>RCL</b> <b>-</b> 변수 $x -$ 변수를 반환합니다.	3-7	
RCLx 변수	<b>RCL</b> <b>x</b> 변수 $x \times$ 변수를 반환합니다.	3-7	
RCL÷ 변수	<b>RCL</b> <b>÷</b> 변수 $x \div$ 변수를 반환합니다.	3-7	
RMDR	<b>←</b> <b>INTG</b> <b>3</b> ( <b>3Rmdr</b> ) 두 정수를 사용한 나눗셈 연산의 나머지를 구합니다.	6-16	1
RND	<b>←</b> <b>RND</b> 반올림 FIX $n$ 디스플레이 모드에서 $x$ 를 소수 $n$ 자릿수로 반올림하거나, SCI $n$ 또는 ENG $n$ 디스플레이 모드에서 $n + 1$ 유효 자릿수로 반올림하거나, 분수 디스플레이 모드에서 표시된 분수와 가장 가까운 10진수로 반올림합니다.	4-18 5-8	1
<b>RPN</b>	<b>MODE</b> <b>5</b> ( <b>5RPN</b> ) RPN(역 폴란드 표기법)을 활성화합니다.	1-9	
RTN	<b>←</b> <b>RTN</b> 반환 프로그램의 끝을 표시하고, 프로그램 포인터가 맨 위나 호출 루틴으로 돌아갑니다.	13-4 14-1	































이름	키 및 설명	페이지	*
R↓	<p><b>[R]</b> <i>롤다운</i></p> <p>RPN 모드에서 <i>y</i>를 Z 레지스터로 이동하고, <i>z</i>를 Y 레지스터로 이동하고, <i>y</i>를 X 레지스터로 이동하고, <i>x</i>를 T 레지스터로 이동합니다.</p> <p>ALG 모드에서 스택을 검토하는 X,Y,Z,T 메뉴를 표시합니다.</p>	2-3 C-7	
R↑	<p><b>[R]</b> <i>롤업</i></p> <p>RPN 모드에서 <i>y</i>를 X 레지스터로 이동하고, <i>z</i>를 T 레지스터로 이동하고, <i>y</i>를 Z 레지스터로 이동하고, <i>x</i>를 Y 레지스터로 이동합니다.</p> <p>ALG 모드에서 스택을 검토하는 X,Y,Z,T 메뉴를 표시합니다.</p>	2-3 C-7	
<b>[S.σ]</b>	표준 편차 메뉴를 표시합니다.	12-4	
SCI <i>n</i>	<p><b>[DISP]</b> <b>[2]</b> (2SCI) <i>n</i></p> <p>소수 <i>n</i> 자릿수의 공학 디스플레이를 선택합니다. <i>n</i> = 0-11</p>	1-22	
SEED	<p><b>[SEED]</b> 난수 시퀀스를 초기값 <math> x </math>로 다시 시작합니다.</p>	4-15	
SF <i>n</i>	<p><b>[FLGS]</b> <b>[1]</b> (1SF) <i>n</i></p> <p>플래그 <i>n</i> (<i>n</i> = 0-11)을 설정합니다.</p>	14-12	
SGN	<p><b>[INTG]</b> <b>[1]</b> (1SGN) <i>x</i>의 부호를 나타냅니다.</p>	4-17	1
<b>[SHOW]</b>	<i>x</i> 의 전체 가수(12자릿수 전체) 또는 현재 프로그램 줄의 숫자를 표시하고, 방정식과 프로그램의 16진수 체크섬과 10진수 바이트 길이를 표시합니다.	6-19 13-23	

이름	키 및 설명	페이지	*
SIN	<b>[SIN]</b> Sine sin $x$ 를 반환합니다.	4-3	1
SINH	<b>[←] [HYP] [SIN]</b> 쌍곡선 사인 sinh $x$ 를 반환합니다.	4-6	1
<b>[→] SOLVE</b> 변수	<b>[→] [SOLVE]</b> 변수 변수 및 $x$ 의 초기 추정치를 사용하여 FN=에 의해 선택된 프로그램이나 표시된 방정식을 해결합니다.	7-1 15-1	
<b>[→] [SPACE]</b>	방정식 입력 중에 빈 공백 문자를 삽입합니다.	14-14	1
SQ	<b>[→] [x²]</b> 인수의 제곱	6-16	1
SQRT	<b>[√x]</b> $x$ 의 제곱근	6-16	1
STO 변수	<b>[→] [STO]</b> 변수 저장. $x$ 를 변수에 복사합니다.	3-2	
STO + 변수	<b>[→] [STO] [+]</b> 변수 변수 + $x$ 를 변수에 저장합니다.	3-6	
STO - 변수	<b>[→] [STO] [-]</b> 변수 변수 - $x$ 를 변수에 저장합니다.	3-6	
STO × 변수	<b>[→] [STO] [×]</b> 변수 변수 × $x$ 를 변수에 저장합니다.	3-6	
STO ÷ 변수	<b>[→] [STO] [÷]</b> 변수 변수 ÷ $x$ 를 변수에 저장합니다.	3-6	
STOP	<b>[R/S]</b> 실행/중지 현재 프로그램 줄에서 프로그램 실행을 시작하고, 실행 중인 프로그램을 중지하고 X 레지스터를 표시합니다.	13-19	
<b>[→] [SUMS]</b>	합계 메뉴를 표시합니다.	12-4	



이름	키 및 설명	페이지	*
sx	<b>S.D.</b> ( $\Sigma x$ ) x 값의 표본 표준 편차를 반환합니다. $\sqrt{\sum (x_i - \bar{x})^2 \div (n-1)}$	12-6	1
sy	<b>S.D.</b> ( $\Sigma y$ ) y 값의 표본 표준 편차를 반환합니다. $\sqrt{\sum (y_i - \bar{y})^2 \div (n-1)}$	12-6	1
TAN	<b>TAN</b> <i>탄젠트</i> . tan x를 반환합니다.	4-3	1
TANH	<b>HYP</b> <b>TAN</b> <i>쌍곡선 탄젠트</i> tanh x를 반환합니다.	4-6	1
VIEW 변수	<b>VIEW</b> <i>변수</i> 값을 스택으로 다시 호출하지 않고 <i>변수</i> 의 레이블 지정된 내용을 표시 합니다.	3-4 13-15	
<b>XEQ</b>	표시된 방정식을 계산합니다.	6-12	
XEQ 레이블	<b>XEQ</b> <i>레이블</i> <i>레이블</i> 에 의해 식별된 프로그램을 실행합니다.	14-1	
x <sup>2</sup>	<b>x<sup>2</sup></b> x의 제곱	4-2	1
$\sqrt{x}$	<b>√x</b> x의 제곱근	4-2	1
$\sqrt[y]{x}$	<b>√y</b> y의 x 거듭제곱근	4-2	1
$\bar{x}$	<b>x.y</b> ( $\bar{x}$ ) x 값의 평균을 반환합니다. $\Sigma x_i \div n$ .	12-4	1
$\hat{x}$	<b>L.R.</b> ( $\hat{x}$ ) X 레지스터에 y 값이 제공되면 회귀 선을 기반으로 x 추정치를 반환합니 다. $\hat{x} = (y - b) \div m$	12-11	1

이름	키 및 설명	페이지	*
!	  팩토리얼(또는 감마). $(x)(x-1) \dots (2)(1)$ 또는 $\Gamma(x+1)$ 을 반환합니다.	4-15	1
XROOT	  $x_2$ 의 $x_1$ 거듭제곱근	6-16	1
$\bar{x} w$	    $(\bar{x} w)x$ 값의 가중 평균을 반환합니다. $(\sum y_i x_i) \div \sum y_i$	12-4	1
 	평균(산술 평균) 메뉴를 표시합니다.	12-4	
$x \leftrightarrow y$ 변수	  $x$ 교환 $x$ 를 변수와 교환합니다.	3-8	
$x \leftrightarrow y$	  $x$ 와 $y$ 교환 $x$ 를 Y 레지스터로 이동하고 $y$ 를 X 레지스터로 이동합니다.	2-4	
  	" $x?y$ " 비교 테스트 메뉴를 표시합니다.	14-7	
$x \neq y$	   ( $\neq$ ) $x \neq y$ 이면 다음 프로그램 줄을 실행합니다. $x = y$ 이면 다음 프로그램 줄을 건너뛸니다.	14-7	
$x \leq y?$	    ( $\leq$ ) $x \leq y$ 이면 다음 프로그램 줄을 실행합니다. $x > y$ 이면 다음 프로그램 줄을 건너뛸니다.	14-7	
$x < y?$	     ( $<$ ) $x < y$ 이면 다음 프로그램 줄을 실행합니다. $x \geq y$ 이면 다음 프로그램 줄을 건너뛸니다.	14-7	

이름	키 및 설명	페이지	*
$x > y?$	     (>) <p><math>x &gt; y</math>이면 다음 프로그램 줄을 실행합니다.  <math>x \leq y</math>이면 다음 프로그램 줄을 건너뜁니다.</p>	14-7	
$x \geq y?$	       ( $\geq$ ) <p><math>x \geq y</math>이면 다음 프로그램 줄을 실행합니다.  <math>x &lt; y</math>이면 다음 프로그램 줄을 건너뜁니다.</p>	14-7	
$x = y?$	       (=) <p><math>x = y</math>이면 다음 프로그램 줄을 실행합니다.  <math>x \neq y</math>이면 다음 프로그램 줄을 건너뜁니다.</p>	14-7	
 	"x?0" 비교 테스트 메뉴를 표시합니다.	14-7	
$x \neq 0?$	  ( $\neq$ ) <p><math>x \neq 0</math>이면 다음 프로그램 줄을 실행합니다.  <math>x = 0</math>이면 다음 프로그램 줄을 건너뜁니다.</p>	14-7	
$x \leq 0?$	   ( $\leq$ ) <p><math>x \leq 0</math>이면 다음 프로그램 줄을 실행합니다.  <math>x &gt; 0</math>이면 다음 프로그램 줄을 건너뜁니다.</p>	14-7	
$x < 0?$	    (<) <p><math>x &lt; 0</math>이면 다음 프로그램 줄을 실행합니다.  <math>x \geq 0</math>이면 다음 프로그램 줄을 건너뜁니다.</p>	14-7	

이름	키 및 설명	페이지	*
$x > 0?$	$x?0$ $>$ $>$ $>$ ( $>$ ) $x > 0$ 이면 다음 프로그램 줄을 실행합니다. $x \leq 0$ 이면 다음 프로그램 줄을 건너뜁니다.	14-7	
$x \geq 0?$	$x?0$ $>$ $>$ $>$ $>$ ( $\geq$ ) $x \geq 0$ 이면 다음 프로그램 줄을 실행합니다. $x < 0$ 이면 다음 프로그램 줄을 건너뜁니다.	14-7	
$x = 0?$	$x?0$ $>$ $>$ $>$ $>$ $>$ ( $=$ ) $x = 0$ 이면 다음 프로그램 줄을 실행합니다. $x \neq 0$ 이면 다음 프로그램 줄을 건너뜁니다.	14-7	
XOR	LOGIC $2$ ( $2 \times \text{XOR}$ ) 논리 연산자	11-4	1
$xiy$	DISPLAY $9$ ( $9 \times i \cdot y$ ) 복소수 표시를 변경합니다.	4-11	
$x+yi$	DISPLAY $\cdot 1$ ( $11 \times + y \cdot i$ ) 복소수 표시를 변경합니다. ALG 모드에만 해당합니다.	1-25	
$\bar{y}$	$x, y$ $>$ ( $\bar{y}$ ) $y$ 값의 평균을 반환합니다. $\Sigma y_i \div n$	12-4	1
$\hat{y}$	L.R. $>$ ( $\hat{y}$ ) X 레지스터에 $x$ 값이 제공되면 회귀선을 기반으로 $y$ 추정치를 반환합니다. $\hat{y} = mx + b$	12-11	1

이름	키 및 설명	페이지	*
$y^x$	$y^x$ 거듭제곱 $y$ 의 $x$ 제곱을 반환합니다.	4-2	1

**참고:**

1. 방정식에 함수를 사용할 수 있습니다.



# 색인

## 특수 문자

[ FN. 적분 참조

Ł ukasiewicz 2-1


% 함수 4-6


%CHG 인수 4-6, C-3

(i) 14-19, 14-20, 14-21

(l)/(j) 사용 14-21

(j) 14-19

 1-15

 분수 단원의 1-26

$\pi$  4-3, A-2


▲ ▼ 표시기

분수 단원의 5-2, 5-3


← 및 →

2진수 11-8

수식 6-7, 13-8

 백스페이스 키 참조 1-4

· 숫자 입력 커서 참조 1-15

 적분 참조 6-3

 표시기

 표시기 1-1, C-3

/c 값 5-4, B-3, B-4

SOLVE 중지 7-7, 15-1

수식 모드 나가기 6-3, 6-4

적분 중지 8-2, 15-8

프로그램 모드 나가기 13-7

프로그램 일시 중단 13-19

프롬프트 취소 6-14, 13-15

## 숫자

10의 지수 1-15, 1-16

10진수 모드. 밑 모드 참조 A-1

16진수 숫자. 숫자 참조

범위 11-7

변환 11-2

산술 11-4

입력 11-1

1개 변수 통계 12-2

2개 변수 통계 12-2

2의 보수 11-4, 11-6

2진수. 숫자 참조

모든 자릿수 보기 11-8

범위 11-7

변환 11-2

산술 11-4

스크롤링 11-8

입력 11-1

8진수. 숫자 참조

범위 11-7

변환 11-2

산술 11-4

입력 11-1

## 가

가수 1-25

가장 큰 정수 4-18

가중 평균 12-4

각도

단위 변환 4-14

벡터 사이 10-5

암시적 단위 4-4, A-2

형식 변환 4-13

각도 모드 4-4, A-2, B-3

간접 주소 지정 14-19, 14-20, 14-22

값에 대한 프롬프트 6-11, 6-13

거듭제곱 곡선 맞춤 16-1

거듭제곱 함수 1-17, 9-3

계산 6-10, 6-11, 6-12, 7-6, 13-4, 14-2

계산기

관련 질문 A-1

기본값 설정 B-3

대비 조정 1-1

자체 테스트 A-5

작동 테스트 A-4, A-5

재설정 A-4, B-1

접지 A-5

켜기 및 끄기 1-1

환경적 제한 A-2

계산기 재설정 A-4, B-1

계산기 테스트 A-4, A-5

계산기에 대한 도움말 A-1

계산기의 습도 제한 A-2

계승 함수 4-15

곡선 맞춤 12-8, 16-1

곱셈, 나눗셈 10-2

괄호 6-5, 6-6, 6-15  
산술에서 2-5  
수식에서 6-3, 6-9, 6-14  
구문 6-14  
구문(수식) 6-14, 6-19, 13-17  
그래드(각도 단위) 4-4, A-2  
그룹화된 표준 편차 16-18  
극좌표-직교좌표 변환 9-5  
근 7-1  
근 없음 7-7  
근 함수 4-3  
근. SOLVE 참조  
검사 7-7, D-3  
다중 7-9  
수식 7-1  
찾을 수 없음 7-8, D-8  
프로그램 15-1  
프로그램에서 15-6  
기본 상태 14-9  
기수 표시 A-1  
기울기(곡선 맞춤) 6-1, 12-8  
긴 수식 6-7  
길이 6-19, 13-7, B-2  
길이 변환 14-14

## 나

나눗셈의 몫과 나머지 4-2  
난수 4-15, B-4  
내림  
SOLVE D-13  
분수 5-8  
삼각 함수 4-4  
적분 8-6  
통계 12-10  
논리  
AND 11-4  
NAND 11-4  
NOR 11-4  
NOT 11-4  
OR 11-4  
XOR 11-4

## 다

다중 근 7-8  
다항식 13-27  
단위 변환 4-14  
단일 단계 실행 13-12  
대비 조정 1-1  
대출자(재무) 17-1  
도  
각도 단위 4-4  
라디안으로 변환 4-14  
등식 수식 6-9, 6-12, 7-1  
디스플레이  
X 레지스터 표시 2-3  
대비 조정 1-1

## 라

라디안  
각도 단위 4-4, A-2  
도로 변환 4-14  
레지스터라는 2-1  
로그 곡선 맞춤 16-1  
로그 함수 4-1, 9-3, C-5  
루틴  
중첩 14-2, 15-11  
프로그램의 부분 14-1  
호출 14-1  
루프 14-16  
루프 카운터 14-17, 14-21

## 마

마침표(숫자에서) 1-23, A-1  
메뉴  
나가기 1-4  
목록 1-6  
사용 예 1-8  
일반 연산 1-6  
메뉴 키 1-6  
메모리 13-16  
가득참 A-1  
변수 지우기 1-28  
사용 가능한 양 1-28  
사용법 B-1  
수식 지우기 6-9



스택 2-1  
전원을 끄고 유지 관리 1-1  
지우기 1-5, 1-29, A-4, B-1, B-3  
크기 1-28, B-1  
통계 레지스터 지우기 12-2  
프로그램 B-1  
프로그램 지우기 1-29, 13-7,  
13-23, 13-23

메모리 지우기 A-4, B-2

메시지

수식에서 13-17  
요약 F-1  
응답 1-27, F-1  
지우기 1-4  
표시 13-17, 13-19

모집단 표준 편차 12-7

목록. 수식 목록 참조 6-7

무게 변환 4-14

문자 키 1-3

문제 해결 A-4, A-5

미래 잔액(재무) 17-1

밀

기본값 B-3  
변환 11-2  
산술 11-4  
설정 11-1  
표시에 영향 11-6  
프로그램 11-8, 13-25

밀 모드

기본값 B-3  
설정 13-24  
수식 6-5, 6-11, 13-25  
프로그래밍 13-25

## 바

반올림

분수 5-8, 13-18  
숫자 4-18

배터리 1-1, A-3

백스페이스 키

VIEW 취소 3-4  
X 레지스터 지우기 2-3, 2-7  
메뉴 나가기 1-4, 1-8  
메시지 지우기 1-4  
수식 입력 1-4  
연산 1-4

프로그램 줄 삭제 13-20

베셀 함수 8-3

벡터

내적 10-4  
덧셈, 뺄셈 10-1  
두 벡터 사이의 각도 10-5  
변수 또는 레지스터에서 벡터  
만들기 10-7

수식에서 10-6

외적 17-11

좌표 변환 4-12, 9-5

프로그램에서 10-6

변동을 함수 4-6

변수 6-3, 7-1

X와 교환 3-8

간접 주소 지정 14-20, 14-21

내부 산술 3-6

다항식 13-27

모두 지우기 1-5

모든 자릿수 표시 13-16

보기 3-4, 13-16, 13-19

보면서 지우기 13-16

수식에서 6-3, 7-1

수식에서 저장 6-12

숫자 저장 3-1

스택 레지스터 내용 액세스 B-5

스택에서 분리 3-2

이름 3-1

이름 입력 1-3

저장 3-2

적분 8-2, 15-8, C-8

지우기 1-29

카탈로그 1-28, 3-4

풀기 7-1, 15-1, 15-6

풀이 D-1

프로그램 입력 13-15

프로그램 출력 13-16, 13-18

프로그램에서 13-13, 15-1, 15-7

회수 3-2, 3-4

변수 값 저장 6-12

변수 카탈로그 1-28, 3-4

변환

각도 단위 4-13

각도 형식 4-13

길이 단위 4-14

부피 단위 4-14

- 숫자 밑 10-1, 11-1
- 시간 형식 4-13
- 온도 단위 4-14
- 좌표 4-11
- 질량 단위 4-14
- 변환 함수 4-10
- 복소수
  - 보기 9-2
  - 스택 9-2
  - 연산 9-2
  - 인수 값 14-7
  - 입력 9-1
  - 좌표계 9-6
- 부호 값 4-17
- 부호 규칙(재무) 17-1
- 부호(숫자) 1-15, 9-3, 11-6
- 분기 14-2, 14-16, 15-7
- 분모
  - 범위 1-26, 5-2
  - 제어 5-4, 14-10, 14-13
  - 최대값 설정 5-4
  - 함수의 불연속 D-6
- 분수
  - 내림 5-8
  - 및 분모 1-26, 5-4, 14-10, 14-13
  - 및 프로그램 5-10, 13-15, 14-9
  - 반올림 5-8
  - 수식 5-9
  - 약분 5-2, 5-6
  - 입력 1-26
  - 정확도 표시기 5-2, 5-3
  - 통계 레지스터가 아님 5-2
  - 표시 5-2, 5-4, A-2
  - 플래그 14-9
  - 형식 5-6
  - 형식 설정 5-6, 14-10, 14-13
- 분수 표시 모드
  - VIEW에 영향 13-15
  - 반올림에 영향 5-8
  - 설정 5-1, A-2
- 불확실성(적분) 8-2, 8-7
- 비교 테스트 14-7

## 사

- 사용 6-1
- 사인(삼각) 4-4, 9-3, A-2, C-6
- 산술
  - 16진수 11-4
  - 2진수 11-4
  - 8진수 11-4
  - 계산 순서 2-14
  - 긴 계산 2-12
  - 스택 연산 2-5, 9-2
  - 일반 절차 1-18
  - 중간 결과 2-12
- 삼각 함수 4-4, 9-3, C-6
- 상관 계수 12-8, 16-1
- 상수(스택 채우기) 2-7
- 상환(재무) 17-1
- 서브루틴. 루틴 참조 14-1
- 선형 회귀 12-7, 16-1, C-9
- 선형 회귀(추산) 12-8, 16-1
- 설정 14-11
- 소수 생성기 17-7
- 소수부 함수 14-16
- 소수점 A-1
- 수식
  - ALG와 비교 13-4
  - RPN와 비교 13-4
  - 밑 모드 6-5, 6-11, 13-25
  - 삭제 1-5, 6-9
  - 수식 제어 14-10
  - 응용 프로그램 역할 17-1
  - 체크섬 6-9, 13-7, 13-24
  - 편집 1-4
  - 표시 6-6
  - 프로그램에서 삭제 13-20
  - 프로그램에서 표시 13-15, 13-18, 14-10
- 수식 모드
  - 나가기 1-4, 6-3
  - 백스페이스 1-4, 6-8
  - 수식 목록 표시 6-3
  - 시작 6-3, 6-7
  - 프로그램 입력 중 13-7

수식 목록

- EQN 표시기 6-4
- 수식 모드에서 6-3
- 연산 요약 6-3
- 추가 6-4
- 편집 6-8
- 표시 6-6

수학

- 계산 순서 2-14
- 긴 계산 2-12
- 복소수 9-1
- 스택 연산 2-5, 9-2
- 실수 4-1
- 중간 결과 2-2

숫자 6-5

- 숫자 값 6-10, 6-11, 7-1, 7-6, 13-4
- 숫자 부호 변경 1-15, 9-3
- 숫자. 2진수, 16진수 숫자, 8진수, 변수 참조

- 교환 2-4
- 극좌표 직교좌표 변환 4-10
- 내부 표현 11-6
- 마침표 및 쉼표 1-23, A-1
- 모든 자릿수 표시 1-25
- 물리 상수 4-8
- 밑 10-1, 13-26
- 반올림 4-18
- 백분율 함수 4-6
- 버림 11-6
- 범위 1-17, 11-7
- 복소수 9-1
- 부분 찾기 4-17
- 부호 변경 1-15, 9-3
- 분수 1-26, 5-1
- 산술 계산 수행 1-18
- 소수 17-7
- 수식에서 6-9
- 순열 4-15
- 숫자의 A-1, E-2
- 실수 4-1
- 음수 1-15, 9-3, 11-6
- 입력 1-15, 1-16, 11-1
- 재사용 2-6, 2-10
- 저장 3-2
- 전체 자릿수 D-2
- 지우기 1-4, 1-5, 1-17

- 큰 숫자와 작은 숫자 1-15, 1-17
- 편집 1-4, 1-17
- 표시 형식 1-21, 11-6
- 프로그램에서 13-8
- 회수 3-2

- 숫자의 E 1-15, 1-22, A-1
- 쉼표(숫자에서) 1-23, A-1
- 스크롤링 6-7, 13-7, 13-16
- 수식 6-7, 13-8, 13-17
- 스택 롤링 2-3, C-7
- 스택 사용법 6-11
- 스택 올리기 참조

- ENTER**의 영향 2-6
- VIEW의 영향을 받지 않음 13-16
- X 및 Y 교환 2-4

- 검토 2-3
- 긴 계산 2-12
- 레지스터 2-1
- 롤링 2-3
- 목적 2-1, 2-2
- 변수에서 분리 3-2
- 변수와 교환 3-8
- 복소수 9-2
- 상수로 채우기 2-7
- 수식 사용법 13-12
- 연산 2-1, 2-5, 9-2
- 크기 제한 2-4, 9-2
- 프로그램 계산 13-15
- 프로그램 입력 13-13
- 프로그램 출력 13-13
- 프롬프트의 영향을 받음 6-14, 13-14

- 스택 올리기. 스택 참조
- 기본 상태 B-4
- 설정 B-4
- 연산 2-5
- 영향 없음 B-4
- 해제 B-4

- 스택. 스택 올리기 참조
- 검토 C-7
- 롤링 C-7
- 시간 형식 4-13
- 실수

- 연산 4-1
- 실수부(복소수) 9-1
- 쌍곡선 함수 4-6, C-6

## 아

알파 문자 1-3  
역 삼각 함수 4-4, C-6  
역 쌍곡선 함수 4-6  
역 정규 분포 16-12  
역 함수 9-3  
연산 요약 6-3  
연산자 우선 순위 6-14  
연속 메모리 1-1  
오류  
수정 2-8  
지우기 1-4  
오류 메시지 F-1  
오버플로우  
계산 결과 1-17, 11-5  
발생 테스트 14-9  
응답 설정 14-9, F-4  
플래그 14-9, F-4  
온도  
계산기의 제한 A-2  
단위 변환 4-14  
우선 순위(수식 연산자) 6-14  
유형 6-9  
음수 1-15, 9-3, 11-6  
이동 키 1-3  
이동. GTO 참조 14-5  
이자(재무) 17-3  
일시 중지. PSE 참조 13-19  
입력 6-4, 6-8  
입력 커서  
백스페이스 1-4  
의미 1-17  
입력이나 1-1

## 자

자체 테스트(계산기) A-5  
잔액(재무) 17-1  
재무 계산 17-1  
저장 산술 3-6  
적분 8-2  
결과의 불확실성 8-2, 8-6, E-2  
메모리 사용법 8-2  
목적 8-1  
변수 8-2, C-8  
변수 변형 E-8  
사용 8-2, C-8

어려운 함수 E-2, E-7  
작동 방식 E-1  
정확도 8-2, 8-6, E-1  
제약 15-11  
제한 8-2, 15-8, C-8, E-7  
중지 8-2, 15-8, E-7  
표시 형식 8-6, 8-7  
프로그램 계산 15-7  
프로그램에서 15-10  
필요한 시간 8-6, E-8  
적분 제한 8-2, 15-8, C-8  
전원 표시기 1-1, A-3  
전체 자릿수(숫자) 1-25, D-13  
절대값(실수) 4-17  
절편(곡선 맞춤) 12-8, 16-1  
정규 분포 16-11  
조건부 테스트 14-6, 14-7, 14-9, 14-12, 14-16  
조합 4-15  
좌표  
변환 4-10  
주소 지정  
간접 14-19, 14-20, 14-22  
중간 결과 2-2  
중첩된 루틴 14-2, 15-11  
지수 곡선 맞춤 16-1  
지수 함수 1-16, 4-1, 9-3, C-5  
지우기  
X 레지스터 2-3, 2-7  
메모리 1-29, A-1  
변수 1-28  
수식 6-9  
숫자 1-17  
일반 정보 1-4  
통계 레지스터 12-2  
프로그램 1-19, 13-23  
지정 수식 6-9, 6-11, 6-12, 7-1  
직교좌표-극좌표 변환 4-10, 9-6  
질량 변환 4-14  
질문 A-1  
질문에 대한 답변 A-1

## 차

차용자(재무) 17-1  
참이면 수행 14-6, 15-6  
창(2진수) 11-8

체크섬  
수식 13-17, 13-24  
프로그램 13-22  
초기값(난수) 4-15  
추산(통계) 12-8, 16-1  
추정값(SOLVE용) 7-2, 7-7, 7-8, 7-11, 15-6

## 카

카탈로그  
나가기 1-4, 13-22  
변수 1-28, 3-4  
사용 1-28  
프로그램 1-28  
켜기 및 끄기 1-1  
코사인(삼각) 4-4, 9-3, C-6  
키  
문자 1-3  
알파 1-3  
이동 1-3

## 타

탄젠트(삼각) 4-4, 9-3, A-2, C-6  
테스트 14-9  
테스트 메뉴 14-7  
통계  
1개 변수 데이터 12-2  
2개 변수 데이터 12-2  
계산 12-4  
곡선 맞춤 12-8, 16-1  
그룹화된 데이터 16-18  
분포 16-11  
연산 12-1  
통계 데이터. 통계 레지스터 참조  
1개 변수 12-2  
2개 변수 12-2  
변수의 합계 12-11  
수정 12-2  
입력 12-1  
전체 자릿수 12-10  
지우기 1-5, 12-2  
초기화 12-3  
통계 레지스터. 통계 데이터 참조  
데이터 수정 12-2  
보기 12-11  
분수 없음 5-2

액세스 12-12  
지우기 1-5, 12-2  
초기화 12-2  
합계 포함 12-1, 12-11, 12-12  
통계 메뉴 12-1, 12-4  
통계 변수의 합계 12-11

## 파

평균 메뉴 12-4  
평균(통계)  
계산 12-4  
정규 분포 16-11  
표본 표준 편차 12-6  
표시 형식  
기본값 B-3  
마침표 및 쉼표 1-23, A-1  
반올림에 영향 4-18  
설정 A-1  
적분에 영향 8-3, 8-6, 8-7  
표시기 1-1  
목록 1-13  
배터리 1-1, A-3  
배터리 부족 1-1, A-3  
알파 1-3  
이동 키 1-2 1-2  
플래그 14-22  
표준 편차  
계산 12-6, 12-7  
그룹화된 데이터 16-18  
분포 16-11  
표준 편차 메뉴 12-6, 12-7  
표현식 수식 6-10, 6-11, 7-1  
폴이 7-1, D-6  
프로그램 레이블  
간접 주소 지정 14-20, 14-21, 14-22  
목적 13-5  
보기 13-23  
분기 14-2, 14-4, 14-16  
실행 13-11  
이동 13-23  
입력 13-5, 13-7  
중복 13-7  
지우기 13-7  
체크섬 13-24  
프로그램 실행 13-10

프로그램 입력 모드 1-4, 13-7  
 프로그램 포인터 13-7, 13-12, 13-20,  
 13-22, B-4  
 프로그램. 프로그램 레이블 참조  
   RPN 연산 13-4  
   SOLVE 사용 15-6  
   SOLVE용 15-1, D-1  
   간접 주소 지정 14-20, 14-21,  
     14-23  
   계산 13-4  
   기술 13-12, 14-1  
   긴 숫자 표시 13-8  
   길이 13-23, 13-24, B-1  
   끝에서 리턴 13-5  
   단계별 이동 13-12  
   데이터 입력 13-6, 13-14, 13-15  
   데이터 출력 13-6, 13-15, 13-19  
   데이터에 대한 프롬프트 13-13  
   디자인 13-4, 14-1  
   루틴 14-1  
   루프 14-16, 14-17  
   루프 카운터 14-18  
   메모리 사용법 13-23  
   메시지 13-17, 13-19  
   모두 삭제 1-5  
   모두 지우기 13-24  
   목적 13-1  
   밀 모드 13-26  
   변수 15-1, 15-7  
   분기 14-2, 14-4, 14-6, 14-16  
   분수 5-8, 13-16, 14-9  
   비교 테스트 14-7  
   삭제 1-28  
   수식 13-5, 13-8  
   수식 계산 14-11  
   수식 삭제 13-8, 13-21  
   수식 편집 13-8, 13-21  
   수식 프롬프트 14-11  
   숫자 13-8  
   실행 13-11, 13-21  
   오류 13-20  
   이동 13-12  
   일시 중단 13-20  
   일시 중지 13-20  
   입력 13-7  
   재개 13-17

적분 사용 15-10  
 적분용 15-7  
 조건부 테스트 14-7, 14-9,  
     14-12, 14-17, 15-6  
 줄 삭제 13-21  
 줄 삽입 13-7, 13-21  
 중지 13-15, 13-17, 13-20  
 중지하지 않음 13-19  
 지우기 13-7, 13-23, 13-24  
 체크섬 13-23, 13-24, B-1  
 카탈로그 1-28, 13-23  
 편집 1-4, 13-8, 13-21  
 플래그 14-9, 14-12  
   함수가 허용되지 않음 13-25  
   호출 루틴 14-1, 14-2  
 프로그램에 입력 13-7  
 프로그램에서 13-4, 13-7, 13-24,  
 14-10  
 프로그램에서 편집 13-7, 13-20  
 프로그램에서 프롬프트 14-11, 15-1,  
 15-8  
 프롬프트  
   INPUT 13-13, 13-15, 15-2, 15-8  
   수식 6-14  
   숨겨진 자릿수 표시 6-14  
   스택에 영향 6-14, 13-15  
   응답 6-13, 13-15  
   지우기 1-4, 6-14, 13-16  
   프로그램된 수식 14-11, 15-1,  
     15-9  
 플래그  
   분수 표시 14-10  
   수식 계산 14-10  
   연산 14-11  
   오버플로우 14-9  
   의미 14-9  
   지우기 14-12  
   지정되지 않음 14-9  
   테스트 14-11  
   표시기 14-11  
   프롬프트 14-11  
**하**  
 함수 6-5, 6-16, G-1  
   2개 인수 1-19  
   단일 인수 1-18, 2-9

목록 G-1  
수식에서 6-5  
실수 4-1  
표시 이름 13-8  
프로그램밍할 수 없음 13-24

함수의 극 D-6  
함수의 최대값 D-8  
함수의 최소값 D-8  
허수부(복소수) 9-1, C-6  
현금 흐름 17-1  
현재 가치. 재무 계산 참조 17-1  
호너의 방법(Horner's method) 13-26  
화폐(재무) 17-1  
화폐의 시간적 가치 17-1  
확률  
이름 입력 1-3  
정규 분포 16-11  
프로그램 카탈로그 1-28, 13-23  
함수 4-15  
회귀(선형) 12-7, 16-1  
회수 산술 3-7  
흐름 다이어그램 14-2

## A

ALG  
수식과 비교 13-4  
프로그램에서 13-4  
ALL 형식. 표시 형식 참조  
설정 1-23  
수식에서 6-5  
프로그램에서 13-7  
A...Z 표시기 1-3, 3-2, 6-4

## B

BIN 표시기 11-1

## C

**[C]**  
VIEW 취소 3-4  
X 레지스터 지우기 2-3, 2-7  
대비 조정 1-1  
메뉴 나가기 1-4, 1-8  
메시지 지우기 1-4  
연산 1-4  
카탈로그 나가기 1-4  
켜기 및 끄기 1-1

프롬프트 취소 1-4  
CLEAR 메뉴 1-5

## D

DSE 14-17

## E

**[ENTER]**  
수식 계산 6-10, 6-11  
수식 종료 6-4, 6-8, 13-7  
숫자 구분 1-17, 2-6  
숫자 중복 2-6  
스택 연산 2-6  
스택 지우기 2-6  
표시된 변수 복사 13-15

**[E]** (지수) 1-16  
ENG 형식, 표시 형식 참조 1-22  
EQN LIST TOP 6-7, F-2  
EQN 표시기  
수식 목록에서 6-4, 6-7  
프로그램 모드에서 13-7

## F

**[FDISP]**  
표시 모드 전환 5-1, A-2  
프로그램밍할 수 없음 5-9, 5-10  
플래그 전환 14-9  
FIX 형식, 표시 형식 참조 1-21  
FN. 적분 참조 8-2  
FN=  
프로그램 적분 15-8  
프로그램 풀이 15-1  
프로그램에서 15-10

## G

**[GTO]**  
PRGM TOP 찾기 13-6, 13-21,  
14-6  
프로그램 레이블 찾기 13-10,  
13-22, 14-5  
프로그램 줄 찾기 4-15, 13-22,  
14-5  
Grandma Hinkle 12-7  
GTO 14-5

## H

HEX 표시기 11-1

## I

i 3-9

INPUT

SOLVE 프로그램에서 15-2

응답 13-14

적분 프로그램에서 15-8

정수부 함수 4-17

프로그램 데이터 입력 13-12

항상 프롬프트 14-11

ISG 14-17

## J

j 3-9, 14-19, 14-20

## L

LAST X 레지스터 2-8, B-5

LASTx 함수 2-8

## M

**MEM**

메모리 검토 1-28

변수 카탈로그 1-28

프로그램 카탈로그 1-28, 13-22

MEMORY CLEAR A-4, B-3, F-4

MEMORY FULL B-1, F-3

MODES 메뉴

각도 모드 4-4

## O

**OFF** 1-1

OCT 표시기 11-1, 11-4

## P

PRGM TOP 13-5, 13-8, 13-22, F-4

PSE

프로그램 일시 중지 15-10

프로그램 중지 금지 14-11

## R

**R/S**

SOLVE 중지 7-8, 15-1

적분 중지 8-2, 15-8

프로그램 실행 13-23

프로그램 일시 중단 13-20

프로그램 재개 13-17, 13-20

프롬프트 종료 6-11, 6-14, 7-2,  
13-16

**R↓** 및 **R↑** 2-3

RCL 3-2, 13-16

RCL 산술 3-7

RPN

수식과 비교 13-5

원래 2-1

프로그램 실행 13-11

프로그램에서 13-5

## S

**SHOW**

수식 길이 6-19, B-2

수식 체크섬 6-19, B-2

숫자 자릿수 1-25, B-8

프로그램 길이 13-23, B-2

프로그램 체크섬 13-23, B-2

프롬프트 자릿수 6-14

**SPACE** 14-14

SCI 형식. 표시 형식 참조

설정 1-22

프로그램에서 13-8

SOLVE

결과 검사 7-7, D-3

극 D-6

내를 찾을 수 없음 7-8, 15-6, D-8

내림 D-13

다중 근 7-9

목적 7-1

불연속 D-6

사용 7-1

수식 계산 7-1, 7-7

스택 결과 7-2, 7-7, D-3

작동 방식 7-7, D-1

재개 15-1

제약 없음 15-11

중지 7-2, 7-8

초기 추정값 7-2, 7-7, 7-8, 7-12,  
15-6

최소값 또는 최대값 D-8

평평한 영역 D-8

프로그램 계산 15-2

프로그램에서 15-6

STO 3-2, 13-13



STO 산술 3-2  
STOP 13-20

## T

T 레지스터 2-5  
TVM 17-1  
TVM 수식 17-1

## U

YFN, 적분 참조  
% 함수 4-6  
표시기 1-3

## V

VIEW

변수 표시 3-4  
부피 변환 4-14  
스택 영향 없음 13-16  
프로그램 데이터 표시 13-16,  
13-19, 15-6  
프로그램 중지 13-16

## X

**XEQ**

수식 계산 6-10, 6-12  
프로그램 실행 13-11, 13-23  
X ROOT 인수 6-17  
X 레지스터  
VIEW의 영향을 받지 않음 13-16  
Y와 교환 2-4  
변수를 사용한 산술 3-6  
변수와 교환 3-8  
스택의 부분 2-1  
지우기 1-5, 2-3, 2-7  
지우지 않음 2-5  
테스트 14-7  
표시됨 2-3  
프로그램 일시 중지 중 13-20  
프로그램에서 지우기 13-8  
프롬프트의 영향을 받음 6-14

