

개정3판

Visual  
Studio  
2017

쉽게 풀어쓴

C언어  
EXPRESS



천인국 지음

## 제1장 프로그래밍의 개념





# 이번 장에서 학습할 내용



- 프로그래밍 개념
- 프로그래밍 언어
- 알고리즘





# 프로그램이란?

- 컴퓨터 = 하드웨어 + 소프트웨어(프로그램)
- 컴퓨터를 범용적으로 만드는 것은 바로 프로그램



동영상 재생 프로그램



mp3 재생 프로그램



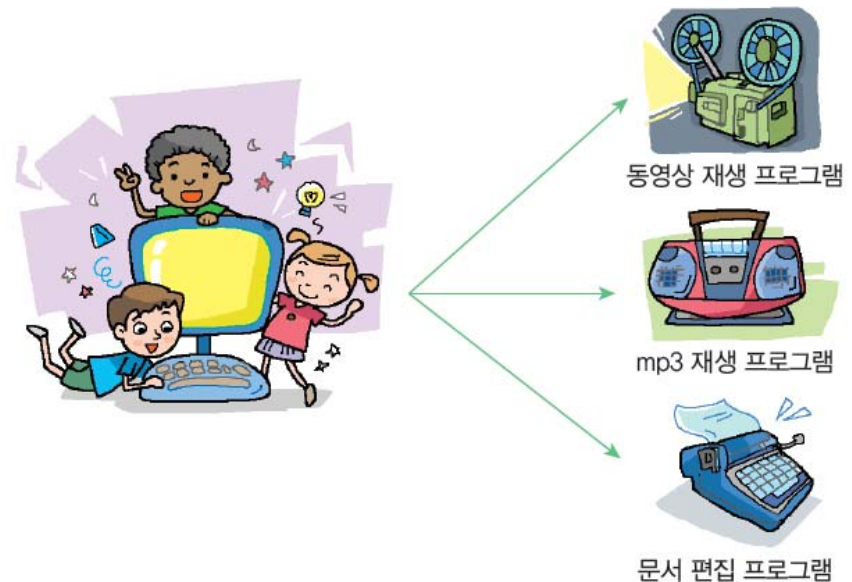
문서 편집 프로그램



# 프로그램

Q) 왜 컴퓨터에서는 가전제품처럼 프로그램 설치 없이 바로 동작되도록 하지 않고 불편하게 사용자가 프로그램을 설치하게 하였을까 ?

A) 컴퓨터를 범용적인 기계로 만들기 위해서이다.  
컴퓨터는 프로그램만 바꾸어주면 다양한 작업을 할 수 있다.





# 계산기와 컴퓨터의 차이

계산기는 정해진 기능만을  
수행한다. 기능을 변경할  
수 없다.



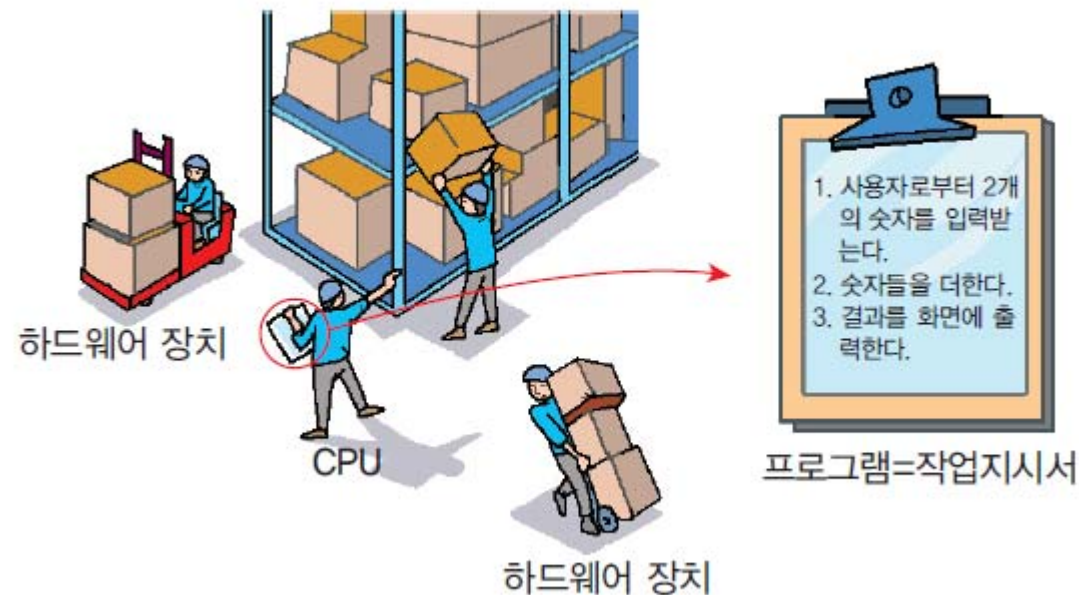
프로그램이라는 개념을  
도입하여 수행하는 기능을  
쉽게 변경할 수 있다.





# 컴퓨터의 정의

- 컴퓨터(computer)는 단순히 계산(compute)만 하는 기계가 아니다.
- 현대적인 의미에서의 컴퓨터는 프로그램(명령어들의 리스트)에 따라 데이터를 처리하는 기계라고 할 수 있다





# 스마트폰도 컴퓨터의 일종

- 피쳐폰: 미리 설정된 기능만 가능
- 스마트폰: 애플리케이션만 변경하면 다양한 용도로 사용 가능





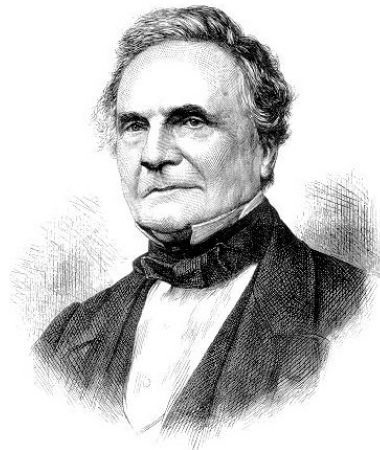


# 프로그램의 역사

- 프로그래밍이 가능한 최초의 기계: 해석 기관(Analytical Engine)

- 만드인이: 찰스 배비지

- 수천 개의 기어, 바퀴, 축, 레버 등이 증기로 작동







# 최초의 프로그래머

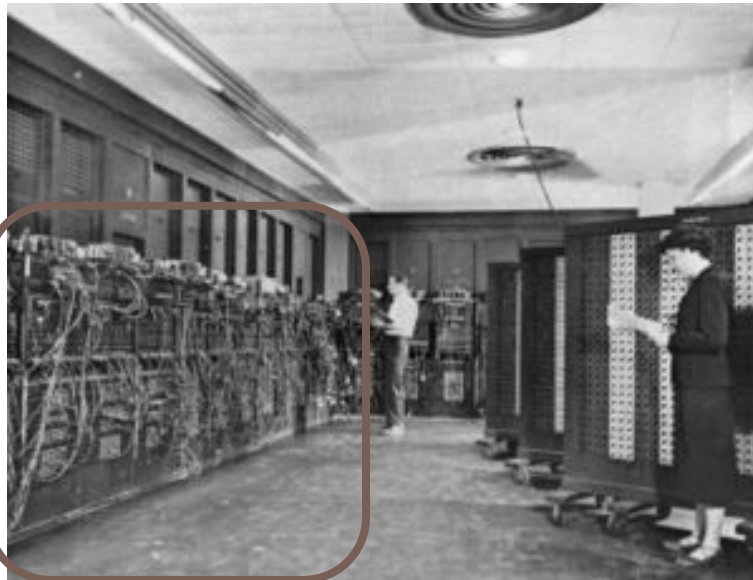
- 프로그램을 최초로 만든 사람은 **에이다 러브레이스(Ada Lovelace)**
- 배비지의 해석 기관에 매료되어 해석 기관을 위한 프로그램을 개발
- 서브루틴(subroutine), 루프(loop), 점프(jump) 등의 핵심적인 컴퓨터 프로그래밍 기본 원리를 고안





# 초기 컴퓨터의 프로그래밍

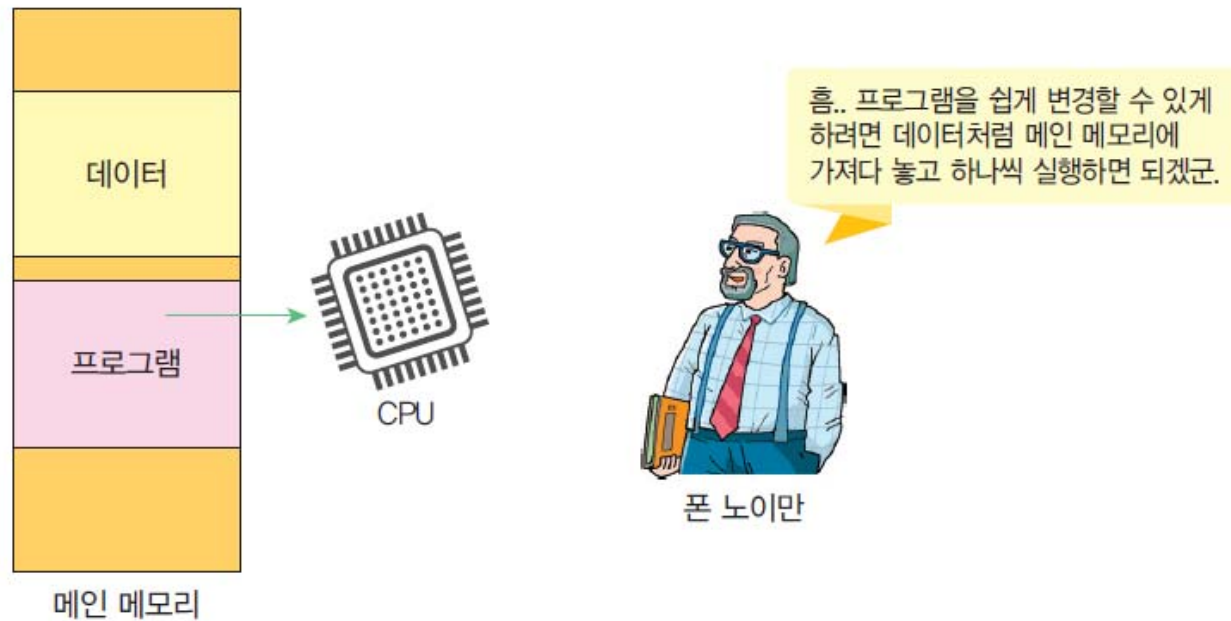
- 초기 컴퓨터인 ENIAC의 프로그램은 스위치에 의하여 기억되었고 프로그램을 변경할 때마다 그 많은 스위치들을 처음부터 다시 연결하여야 했다.





# 폰노이만 구조 (프로그램 내장 구조)

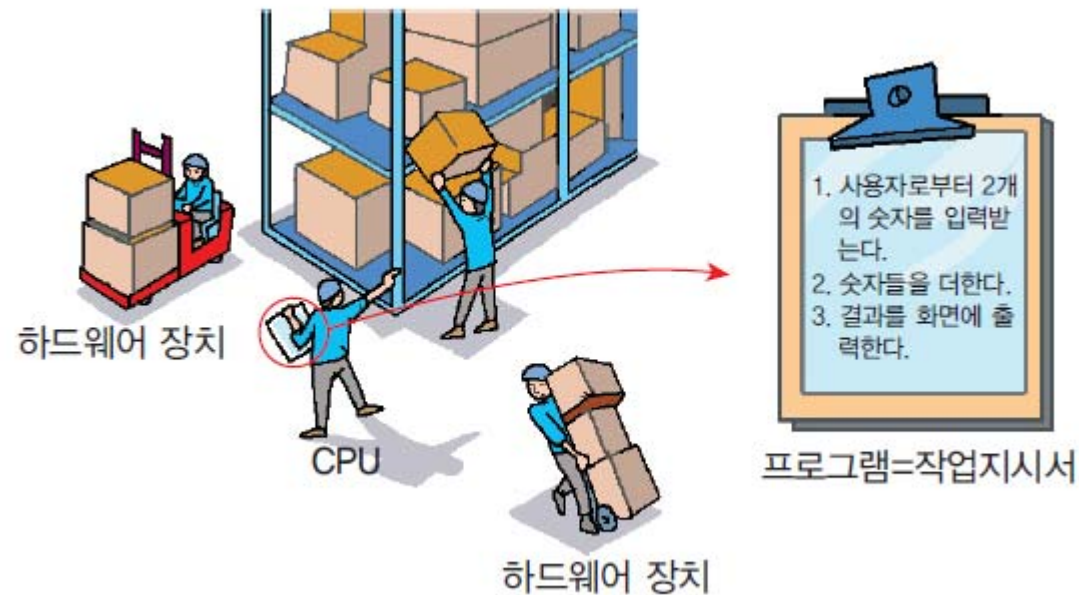
- 프로그램을 **메인 메모리에 저장**
- 메인 메모리에 저장된 프로그램에서 명령어들을 순차적으로 가져와서 실행





# 프로그램==작업지시서

- 프로그램: 컴퓨터에게 해야 할 작업의 내용을 알려주는 문서

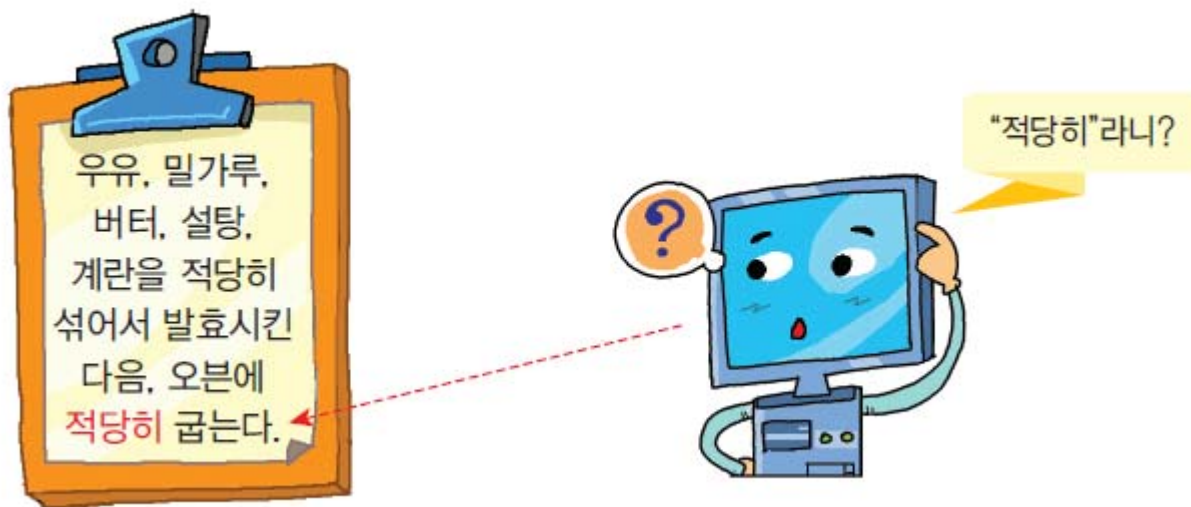


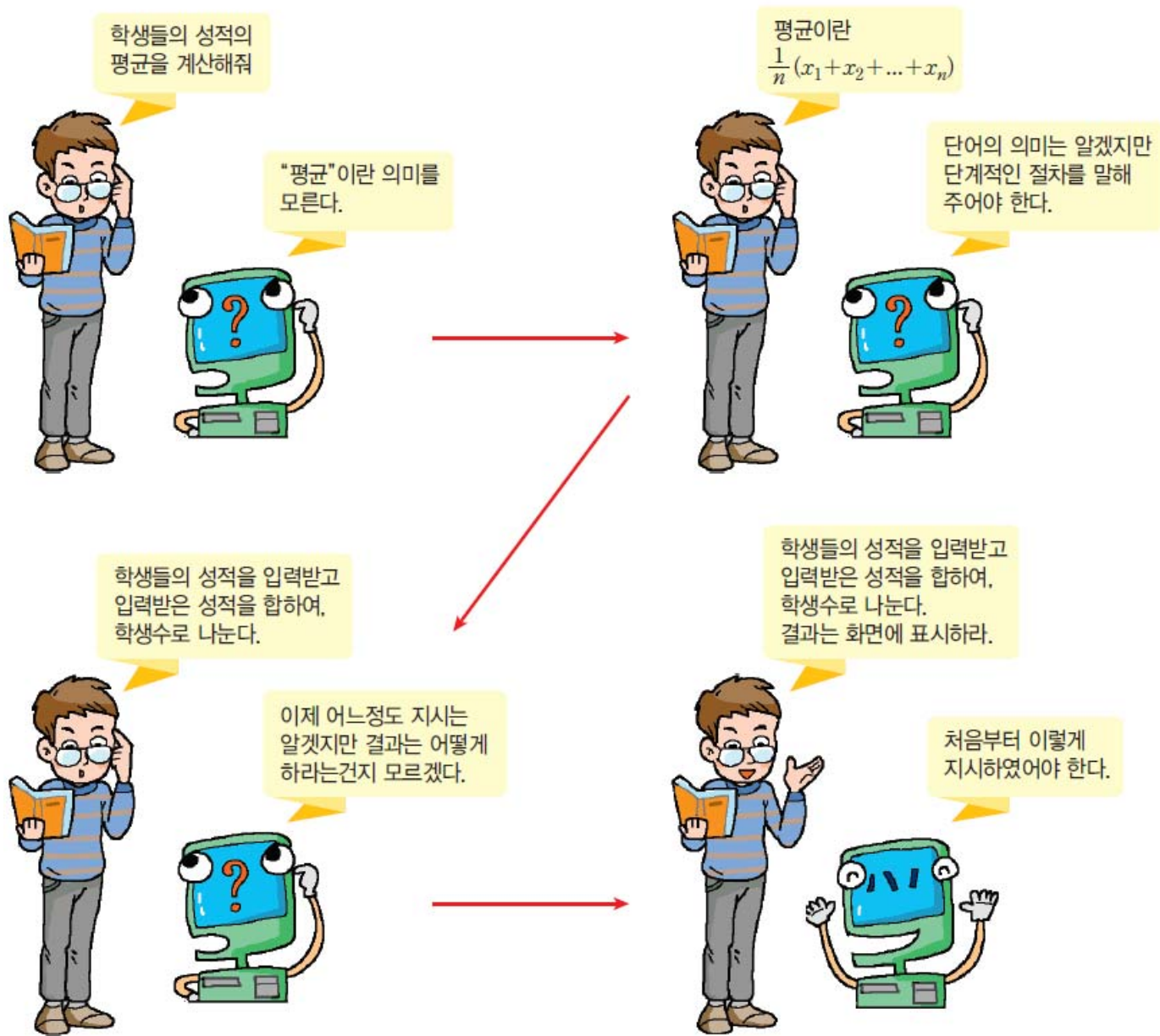


# 작업을 지시하는 방법

Q) 컴퓨터에게 어떻게 작업을 시킬 수 있을까?

A) 상식이나 지능이 없기 때문에 아주 자세하고 구체적으로 일을 지시하여야 한다.









# 이번 장에서 학습할 내용



- 프로그래밍 개념
- 프로그래밍 언어
- 알고리즘



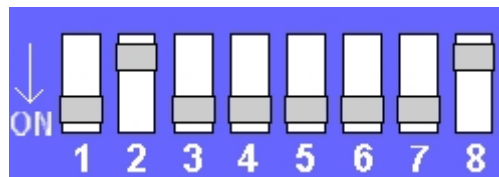


# 기계어

## □ 컴퓨터가 바로 이해할 수 있는 언어는 기계어

컴퓨터가 알아듣는 언어는 한가지로  
0과 1로 구성되어 있는 "001101110001010..."과 같은  
기계어이다.

컴퓨터는 모든 것을 0과 1로 표현하고 0과 1에 의하여  
내부 스위치 회로들이 ON/OFF 상태로 변경되면서 작업을  
한다.





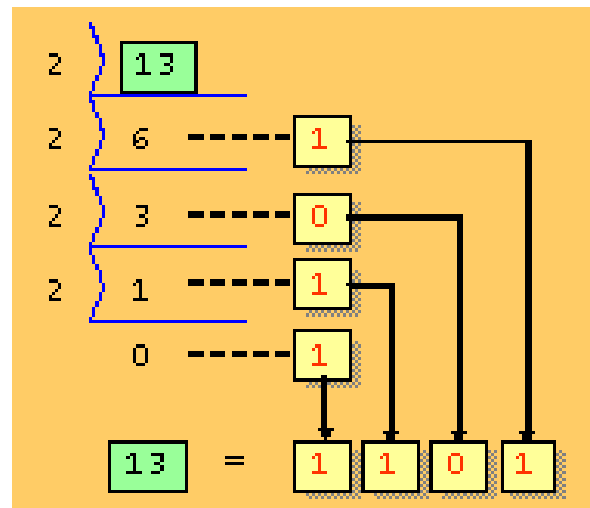
# 이진수

Q) 이진수는 십진수와 무엇이 다른가?

A) 이진수는 0과 1로만 구성되어 있다.

Q) 십진수를 이진수로 바꾸려면?

A) 십진수를 이진수로 바꾸려면 십진수를 2로 나누고 나머지를 기록하는 작업을 몫이 0이 될 때까지 되풀이하면 된다.





# 프로그래밍 언어의 필요성

Q) 그렇다면 인간이 기계어를 사용하면 어떤가?

- 기계어를 사용할 수는 있으나 이진수로 프로그램을 작성하여야 하기 때문에 아주 불편
- 프로그래밍 언어는 자연어와 기계어 중간쯤에 위치

우리말 아니?



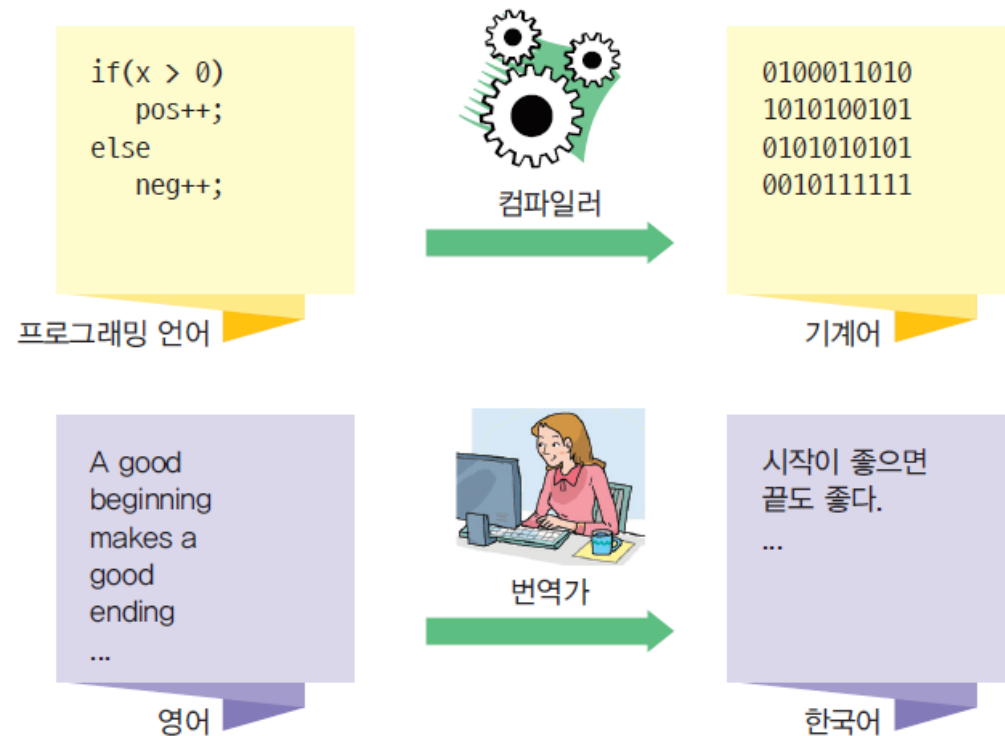
아니 기계어 밖에 몰라





# 컴파일러

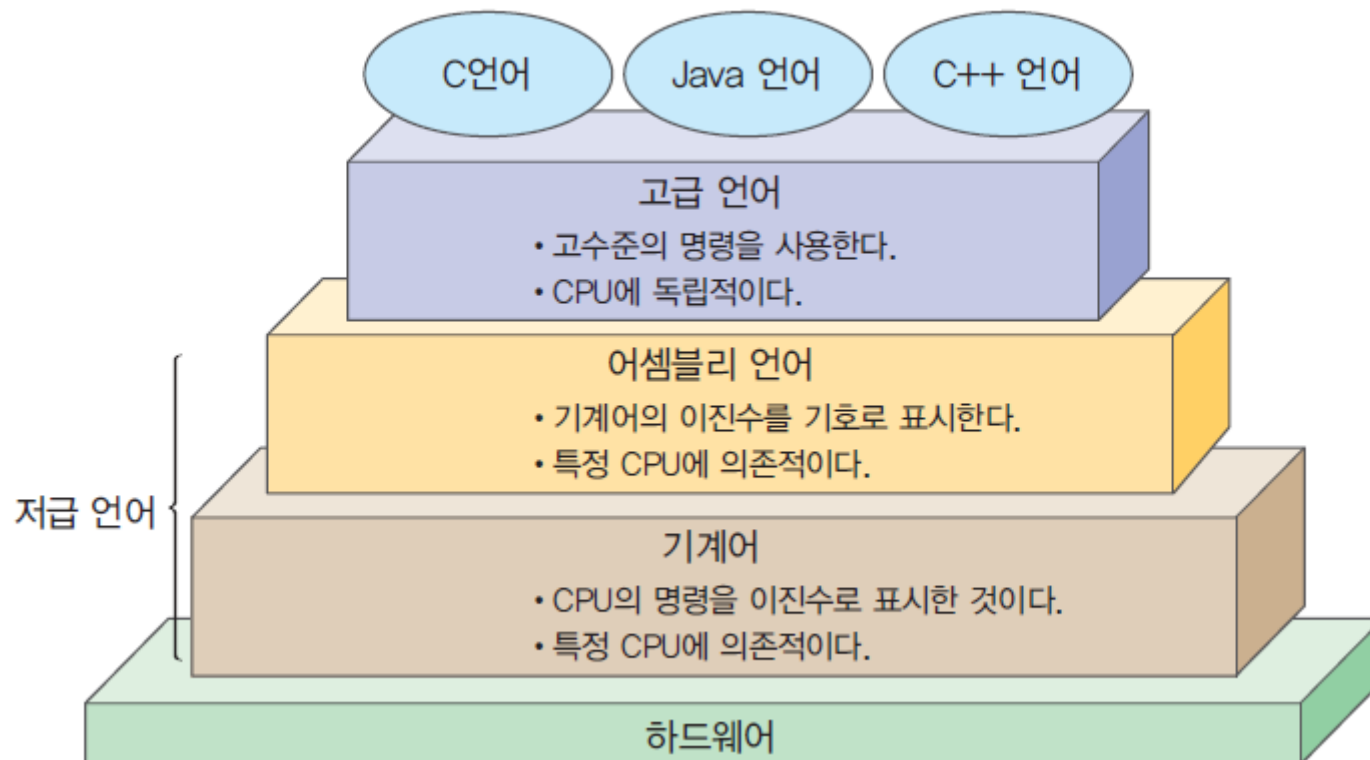
- 컴파일러(compiler): 인간과 컴퓨터 사이의 통역 역할





# 프로그래밍 언어의 분류

- 기계어(machine language)
- 어셈블리어(assembly language)
- 고급 언어(high-level language)







# 기계어

- 특정 컴퓨터의 명령어(instruction)를 이진수로 표시한 것
- 0과 1로 구성
- 하드웨어에 종속 (CPU에서 사용하는 instruction set에 따라 달라짐)

```
00001111 10111111 01000101 11111000
00001111 10111111 01001101 11111000
00000011 10100001
01100110 10001001 01000101 11111010
```



# 어셈블리어

- CPU의 명령어들을 이진수가 아닌 영어의 약자인 기호로 표기
  - 기계어보다는 더 높은 수준에서 프로그램을 작성하는 것 가능
  - 기호와 CPU의 명령어가 일대일 대응
- 어셈블러(assembly): 기호(어셈블리어)를 이진수로 변환하는 프로그램

```
MOV AX, MIDSORE
MOV CX, FINALSORE
ADD AX CX
MOV TOTALSCORE, AX
```



# 고급 언어

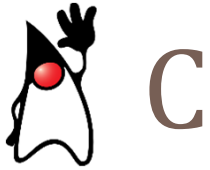
- 특정한 컴퓨터의 구조나 프로세서에 무관하게, 독립적으로 프로그램을 작성할 수 있는 언어

```
TotalScore = MidScore + FinalScore;
```

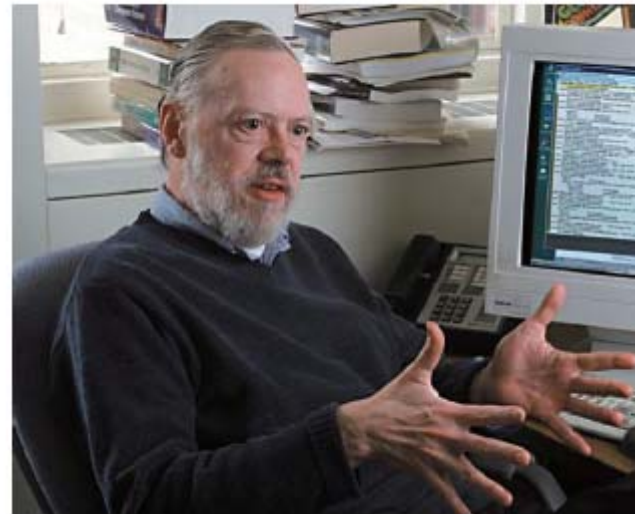
- 고급 언어 종류

C, C++, JAVA, Python, C#, Kotlin, Objective-C, FORTRAN, COBOL, PASCAL

- 컴파일러: 고급 언어 문장을 기계어로 변환하는 프로그램



- 1970년대 초 AT&T의 Dennis Ritchie 에 의하여 개발
- B언어 → C언어
- UNIX 운영 체제 개발에 필요해서 만들어짐
- 처음부터 전문가용 언어로 출발





# C언어의 버전

## □ K&R C

- 1978년 "C Programming Language" 책 출간
- 비공식적인 명세서 역할

## □ ANSI C

- 1983년 ANSI(American National Standards Institute)는 X3J11이라는 위원회를 만들고 C언어 표준 개발. 1989년 ANSI C 표준 발표

## □ C99

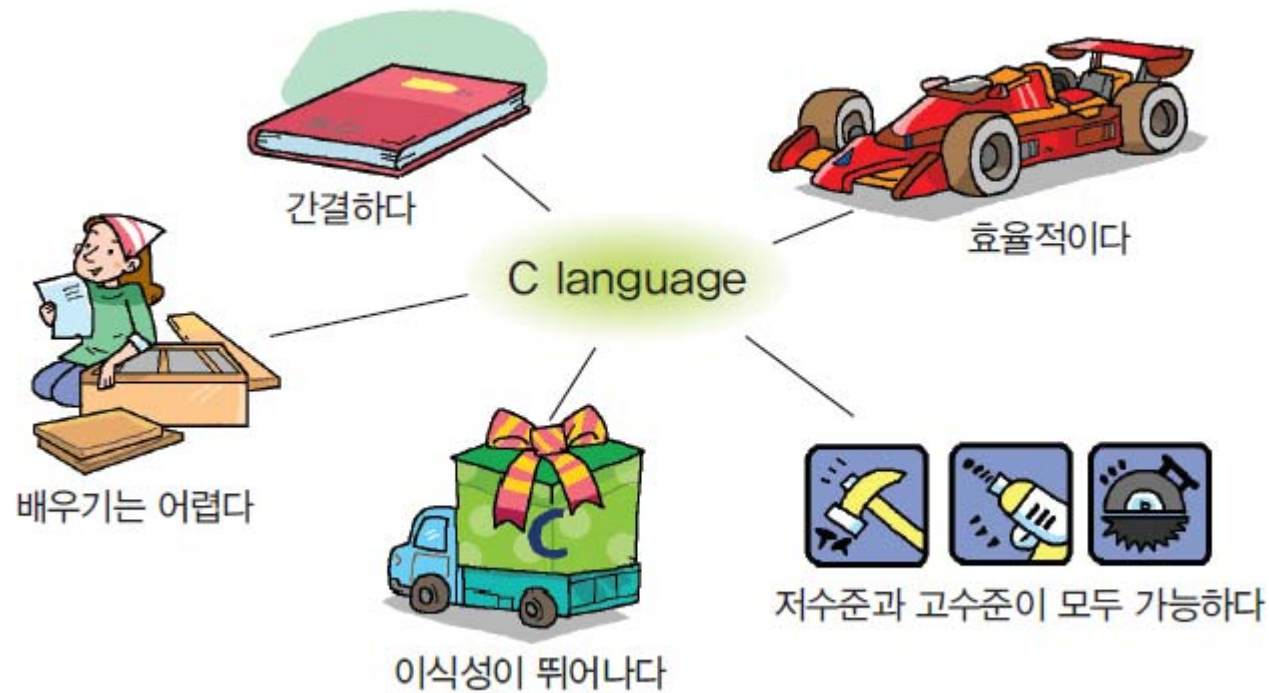
- 1999년에 ISO에 의한 표준
- C++에서 사용되는 특징 추가
- 점차 많은 컴파일러에서 지원

## □ C11

- ISO에 의하여 2011년 12월에 발표된 C언어 표준



# C언어의 특징







# C언어의 미래

## Q) 앞으로도 C언어는 사용될 것인가?

- C언어는 C++와 JAVA의 공통적인 부분이다.
- 임베디드 시스템에서는 C언어가 많이 사용된다.

임베디드 시스템: 임베디드 시스템이란 특수 목적의 시스템으로 컴퓨터가 장치 안에 내장되어 있다는 의미

MP3 플레이어, 핸드폰 등이 여기에 속함



스마트폰도 CPU와 플래시 메모리 등이 들어가 있는 임베디드 시스템이다.



## 중간 점검

- 컴퓨터가 가장 쉽게 이해하는 언어는 무엇인가?
- 컴파일러는 어떤 역할을 하는가?
- 임베디드 시스템이란 무엇인가?
- C언어의 장점과 단점을 정리하여 보자.





# 이번 장에서 학습할 내용



- 프로그램의 이해
- 프로그래밍 언어
- 알고리즘





# 알고리즘

Q) 오븐의 사용법만 배우고 음식 재료만 있으면 누구나 요리가 가능한가?

A) 요리법을 알아야 한다.

- 프로그램이 요리와 같다면 알고리즘은 요리법에 해당
- 알고리즘(algorithm): 문제를 해결하는 절차(방법)
  - 문제를 풀기 위하여 컴퓨터가 수행해야 할 단계적인 절차를 기술한 것
  - 알고리즘을 프로그래밍 언어로 구현하면 프로그램이 됨





# 빵을 만드는 알고리즘

- ① 빈 그릇을 준비한다.
- ② 이스트를 밀가루, 우유에 넣고 저어준다.
- ③ 버터, 설탕, 계란을 추가로 넣고 섞는다.
- ④ 따뜻한 곳에 놓아두어 발효시킨다
- ⑤ 170~180도의 오븐에서 굽는다





# 1부터 10까지의 합을 구하는 알고리즘

- ① 1부터 10까지의 숫자를 직접 하나씩 더한다.

$$1 + 2 + 3 + \dots + 10 = 55$$

- ② 두수의 합이 10이 되도록 숫자들을 그룹핑하여 그룹의 개수에 10을 곱하고 남은 숫자 5를 더한다.

$$\left. \begin{array}{l} (0 + 10) = 10 \\ (1 + 9) = 10 \\ (2 + 8) = 10 \\ (3 + 7) = 10 \\ (4 + 6) = 10 \end{array} \right\}$$

5

$$10 * 5 = 50 \quad + \quad 5 \quad = \quad 55$$

- ③ 공식을 이용하여 계산할 수도 있다.

$$10(1+10)/2=55$$





# 알고리즘의 기술

- 자연어(natural language)
- 의사 코드(pseudo-code)
- 순서도(flowchart)



# 자연어

- 자연어 (natural language)는 인간이 사용하는 언어
- 단어들을 명백하게 정의해야 한다.

1. 리스트의 첫 번째 숫자가 가장 크다고 가정하자.
2. 리스트의 남아있는 숫자들이 하나씩 조사하여 현재의 최대값보다 크면 노트에 적는다.
3. 모든 숫자들을 전부 조사된 후에 노트에 가장 나중에 적힌 숫자가 최대값이 된다.



# 의사 코드

- 의사 코드(**Pseudocode**): 자연어보다는 더 체계적이고 프로그래밍 언어보다는 덜 엄격한 언어로서 알고리즘의 표현에 주로 사용되는 코드

알고리즘 GetLargest

입력: 숫자들의 리스트 L.

출력: 리스트에서 가장 큰 값

```
largest ← L[0]
for each n in L do
  if n > largest then
    largest ← n
return largest
```



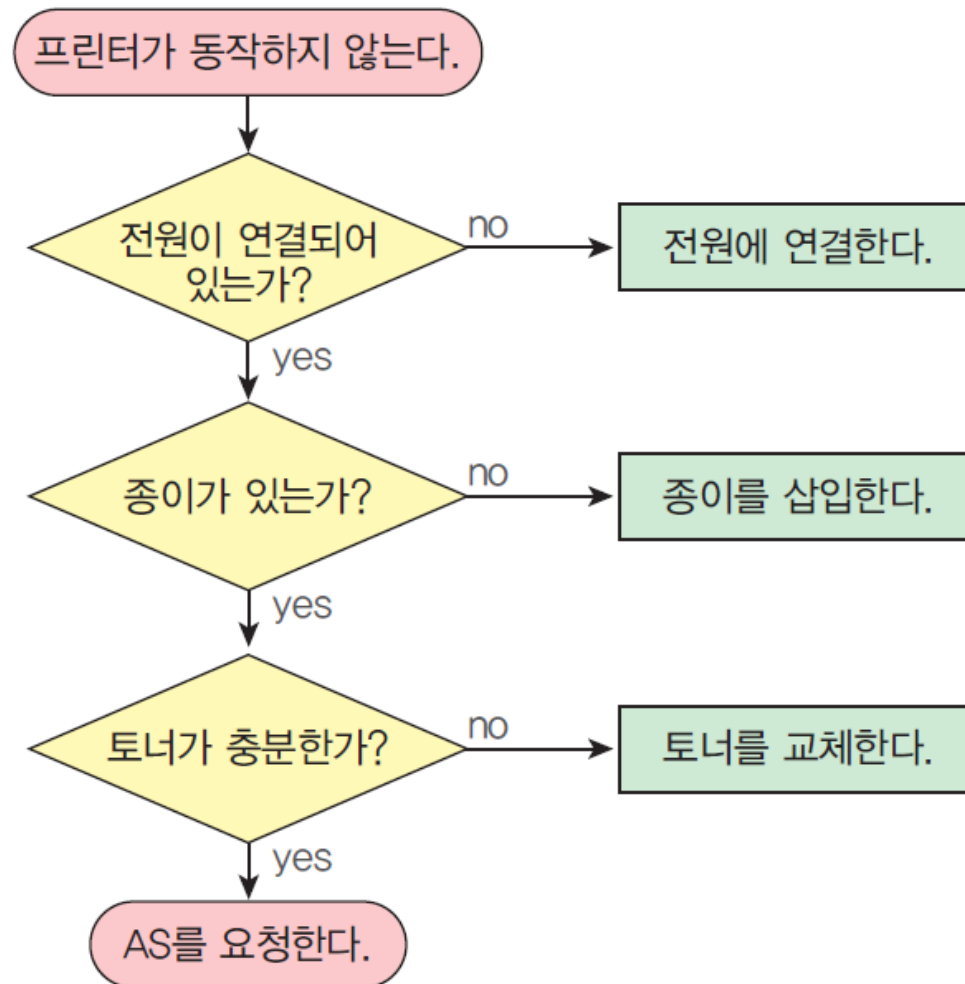
# 순서도

- 순서도(flow chart): 프로그램에서의 논리 순서 또는 작업 순서를 그림으로 표현하는 방법





# 순서도로 표현된 알고리즘의 예





# 알고리즘을 만드는 방법

1. 문제를 한 번에 해결하려고 하지 말고 더 작은 크기의 문제들로 분해한다.
2. 문제가 충분히 작아질 때까지 계속해서 분해한다.

① 방을 청소한다.  
② 거실을 청소한다.  
③ 부엌을 청소한다.

① 환기를 시킨다.  
② 물건들을 정리한다.  
③ 진공 청소기를 돌린다.  
④ 걸레질을 한다.



환기



물건정리



진공청소기



걸레질



## 중간 점검

- 친구에게 전화를 거는 알고리즘을 만들어보자.
- 세탁기를 이용하여서 세탁을 하는 알고리즘을 만들어보자.
- 햄버거 가게에서 햄버거를 주문하는 알고리즘을 순서도로 작성해보자.





# mini project: 3개의 수 중에서 최대 값 찾기

- 사용자로부터 받은 3개의 수 중에서 최대값을 찾는 알고리즘을 순서도로 작성해보자.



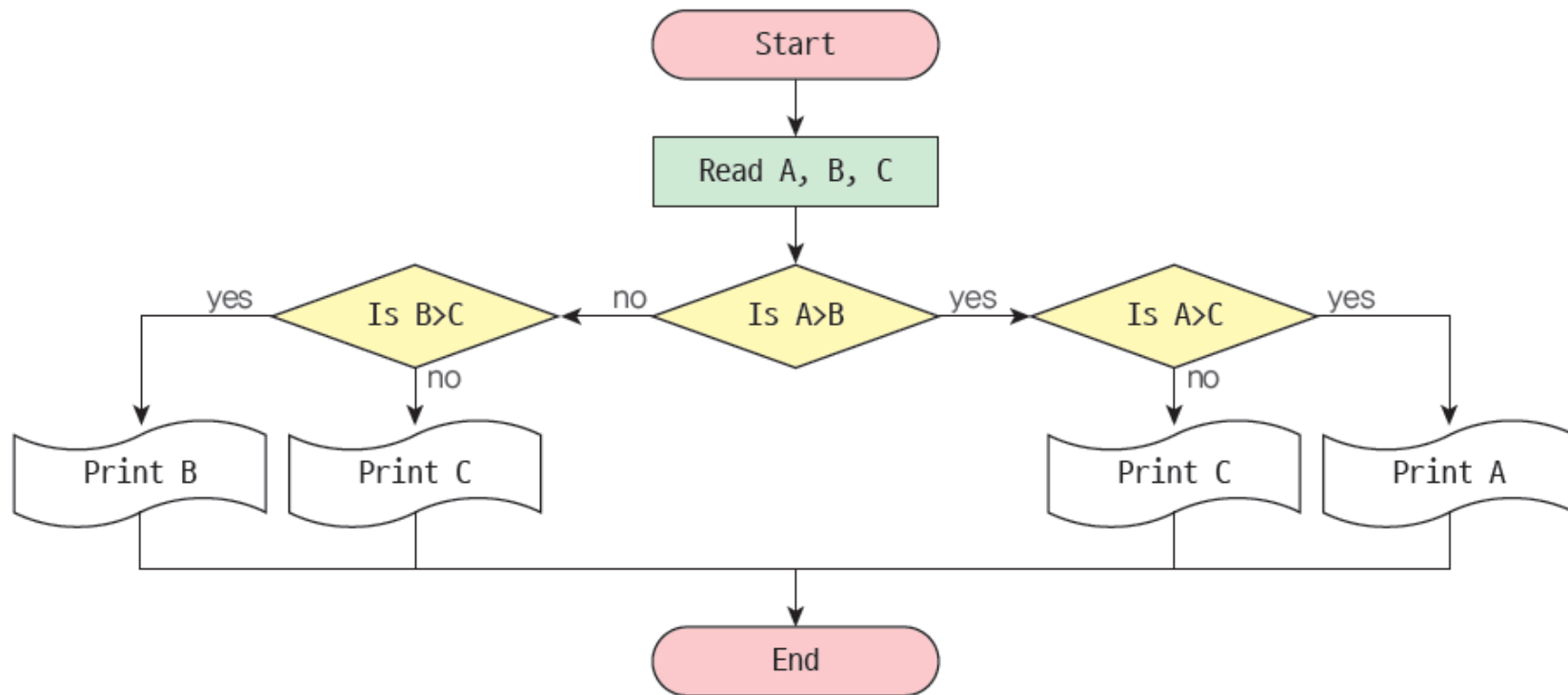


# mini project: 3개의 수 중에서 최대 값 찾기

1. 사용자로부터 받은 3개의 수를 A, B, C라고 하자.
2. 먼저 A와 B를 비교한다.
3. A가 B보다 크면 A와 C를 비교해서 큰 수를 출력한다.
4. 만약 B가 A보다 크다면 B와 C를 비교하여서 큰 수를 출력한다.



# 알고리즘의 순서도 표현





## 도전 문제

- 3개의 수 중에서 최소값을 찾는 알고리즘을 생각해보자.





# Q & A

