

문제 해결 방법 및 절차



GyeongNam Science High School

© copyright 2014 ~ all right reserved by

GTKBS(seotosna@naver.com)

Online Judge Site : <http://www.judgeon.net>

문제 해결 방법 및 절차

- 1. Code::Blocks 소개 1
- 2. 프로그래밍언어의 기초 8
 - 가. 문제해결 방법의 표현 8
 - 나. 프로그래밍언어 시작하기 11
 - 다. 데이터 저장하기 22
 - 라. 입력받고 출력하기 28
 - 마. 파일로 입출력하기 33
 - 바. 계산에 필요한 연산자 38
 - 사. 코드의 흐름에 제어하자 47
 - 아. 반복된 일을 시켜보자 55
 - 자. 대량의 동일 데이터를 저장하는 배열 64
 - 차. 글자는 숫자와 다르다 69
 - 카. 사용자가 직접 만드는 함수 74
 - 타. 여러 가지 데이터를 하나로 구조체 만들기 82
 - 파. 거울속의 거울 86
 - 하. 스택(stack)이 뭐죠? 89
 - 가. 큐(Queue)는 뭐죠? 86
 - 나. 문제해결과정 1 94
 - 다. 문제해결과정 2 96
 - 라. 문제해결과정 3 98
 - 마. 오일러 프로젝트를 시작해 보자 99
- 3. Online Judge Site 소개 101

본 교재는 경남과학고등학교 학생들의 프로그래밍을 통한 문제해결력 함양을 위해 제작되었습니다. 교재의 내용은 문제상황, 입력, 출력을 기준으로 작성하여 온라인 저지 사이트와 연계하여 학습이 가능하도록 구성하여 학생들이 쉽게 접근하도록 구성하였습니다. 또한 통합개발환경에 대해서도 소개를 하여 프로그래밍 툴을 사용하는데 어려움이 없도록 하였다.

1. Code::Blcoks 소개

가. 프로그램 소개

1) Code Blocks개요

CodeBlocks는 GCC를 기본으로 포함한 오픈소스이다. 또한 어떤 환경에서도 사용이 가능하다. 즉, 윈도우, 리눅스에서 설치하여 지원하며 Mac OS까지 지원한다. 개인 또는 기업 등 모두 무료로 사용가능하다. 또한 꾸준히 업데이트가 되고 있다. 교재를 만드는 시점에는 13.12로 되어 있다.

2) 프로그램 설치하기

- 접속 : <http://www.codeblocks.org/downloads/binaries>
- 자신의 환경에 맞게 다운 받으면 된다. 그림은 윈도우 환경에 대한 것이다. 다운 받은 프로그램을 실행하여 설치를 진행하면 된다.

File	Date	Download from
codeblocks-13.12-setup.exe	27 Dec 2013	BerliOS or Sourceforge.net
codeblocks-13.12mingw-setup.exe	27 Dec 2013	BerliOS or Sourceforge.net
codeblocks-13.12mingw-setup-TDM-GCC-481.exe	27 Dec 2013	BerliOS or Sourceforge.net

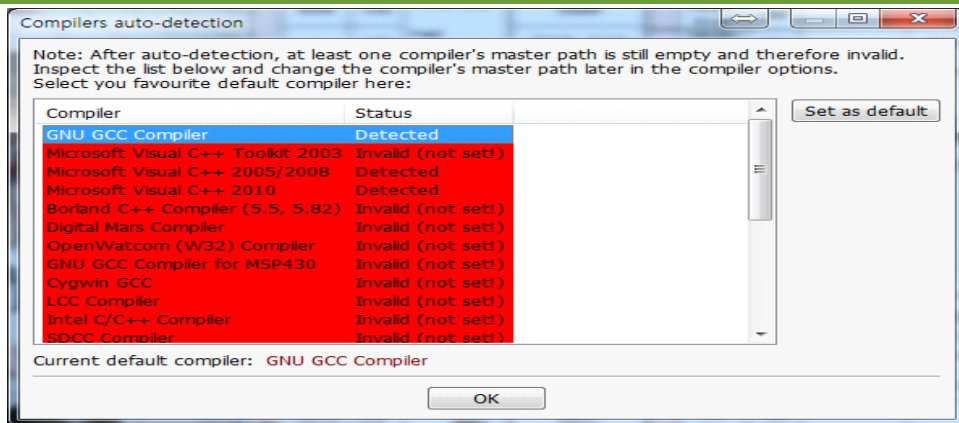


그림. 기본컴파일러 선택

- 설치과정에서 해당 컴파일러가 표시된다. 다운받을 때 GNU GCC Compiler를 포함해서 다운 받았기 때문에 선택하고 Set as default를 선택하면 된다.

3) 프로그램 실행하기

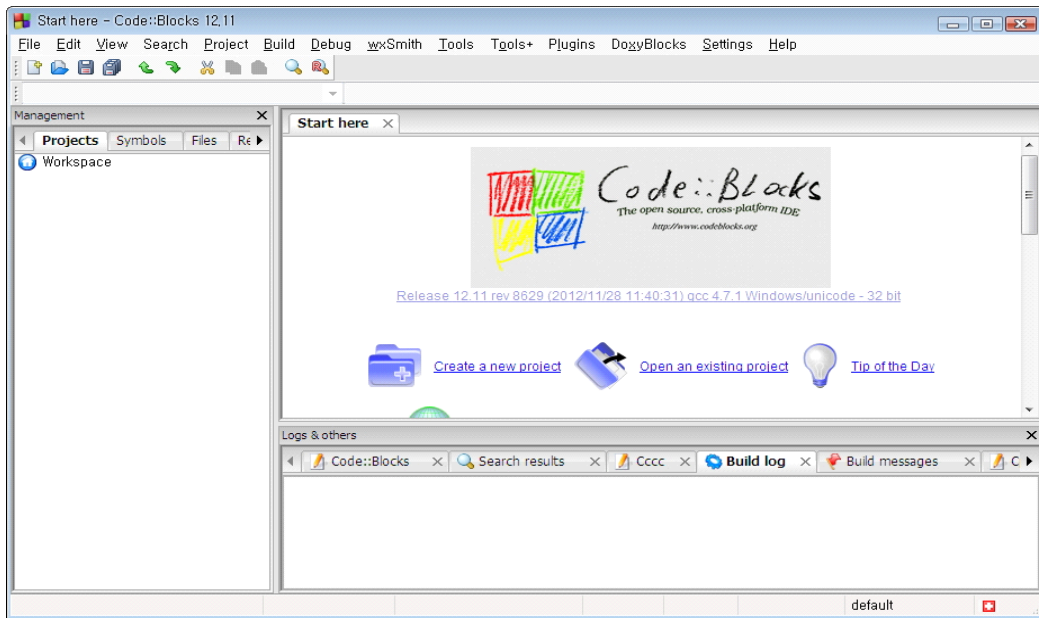


그림. 프로그램 실행화면

4) 프로젝트 생성

프로젝트를 생성하는 이유는 CodeBlocks 자체에서 제공하는 기능을 활용하여 디버깅을 하기 위함이다. 디버깅(Debugging)은 버그를 잡는 것을 말한다.

간단히 이야기 하면 보통 버그는 오류를 말하는 것으로 크게 두 가지의 오류가 있다. 즉 **문법적 오류**와 **논리적 오류**가 있다. 문법적 오류는 프로그래밍언어에서 제공하는 문법적인 측면에서 발생하는 오류이다. 이 오류는 보통 컴파일과정에서 거의 다 찾아준다.

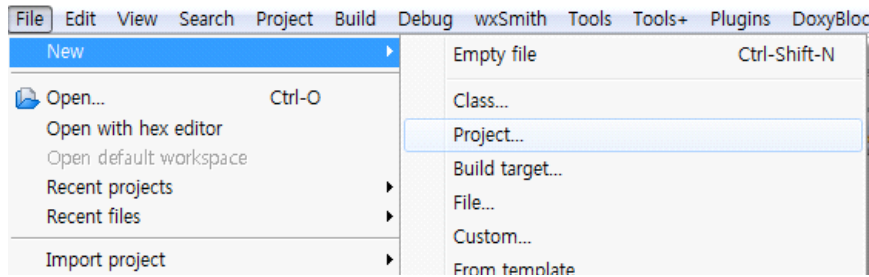
그러나 논리적 오류는 프로그램의 결과가 올바르지 않을 때를 말하는 것이다. 예를 들어 결과가 3이 나와야 하는데 2가 나왔다. 이는 사용자가 프로그램을 잘못 작성하여 틀린 답이 나오는 것으로 문법적인 측면에서는 문제가 없다는 것이다. 이렇게 프로그램을 잘못 작성하여 발생하는 오류를 논리적 오류라고 한다. 이러한 오류는 사실 찾기 힘들기 때문에 통합개발환경에서는 이러한 논리적 오류를 사용자가 찾을 수 있도록 여러 가지 방법을 지원한다.

보통 break point, call stack, memory 주소 등의 정보를 제공하는 것이다. 이를 위해 CodeBlocks에서는 프로젝트를 생성해야만 이용할 수 있다. 실제 이를 이용하는 방법은 실제 수업에서 이해하기 바란다.

먼저 프로젝트를 생성하는 절차는 다음과 같다.

▶ 프로젝트를 안만들고 파일만 만들어 할 경우 ⑧번부터 하면 된다.

① 메뉴에서 [File]-[New]-[Project...]를 선택하여 새 프로젝트를 만듭니다.



② 우리가 만드는 프로그램은 간단한 출력결과를 통해 볼 것이기 때문에 Console application을 선택합니다.

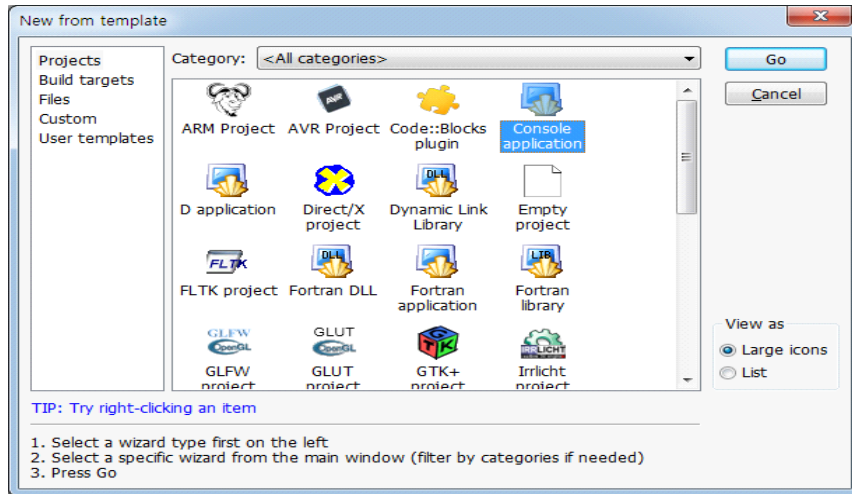


그림. 새로운 프로젝트 생성

③ 첫 화면은 간단한 안내 페이지가 나온다. 별 의미 없으니 무시하고 Next를 선택한다.

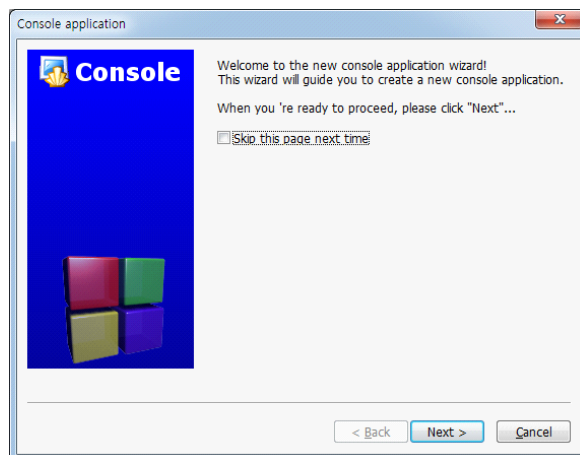


그림. 안내 페이지

④ 언어 선택 창에서 C++를 선택한다. 실제 우리가 다루는 대부분의 내용은 c언어이지만 C++이 C언어를 포함하고 있기 때문에 우리는 C++로 한다. 또한 C++ 처리하면 얻는 이점이 크기 때문이다.

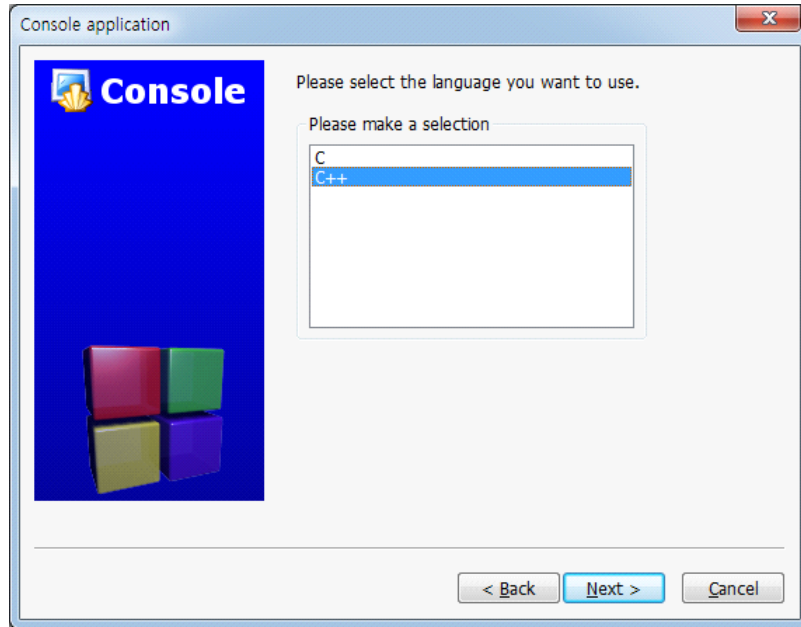


그림. C++선택

⑤ 프로젝트 이름을 지정하고 프로젝트를 생성할 폴더를 지정합니다. 이때 멀티실은 폴더를 만들 때 반드시 D: 에 만들어 지정한다.

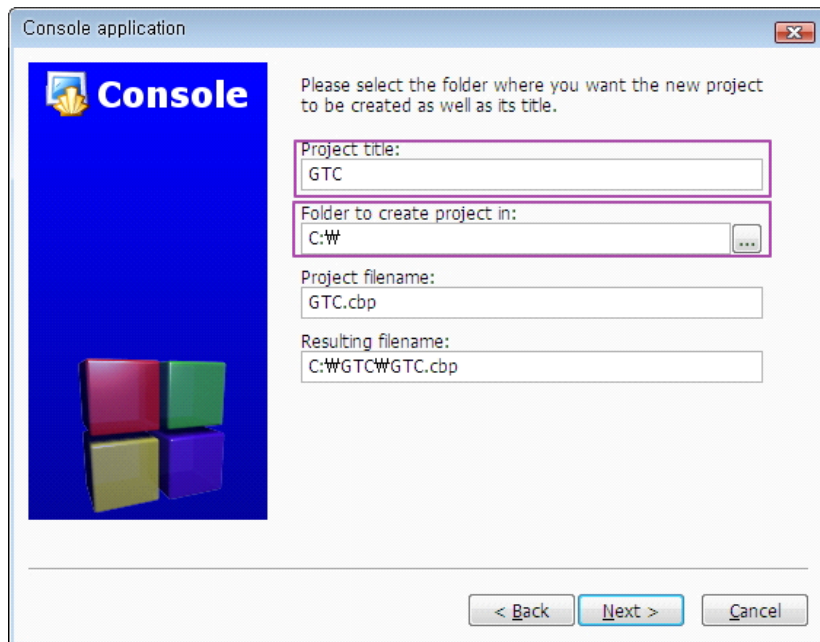


그림. 프로젝트명, 파일명 결정

⑥ 컴파일러 선택 창에서“GNU GCC Compiler”를 선택한 뒤 [Finish] 버튼을 클릭합니다.

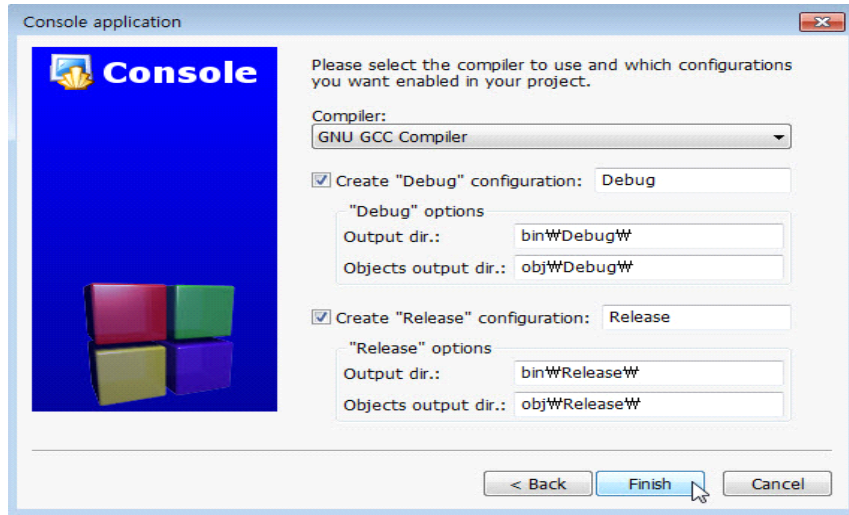


그림. 디버그와 릴리즈 모드 선택

⑦ 그러면 main.cpp 라는 파일이 생성됩니다. main.cpp 이름을 제거합니다.

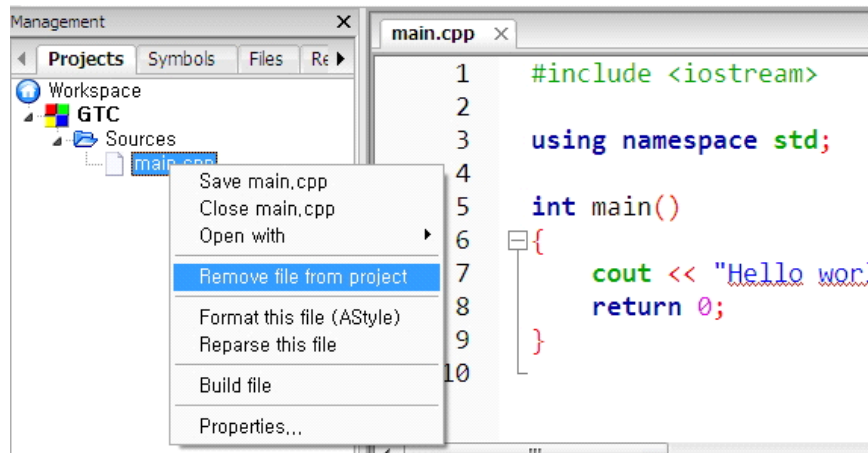
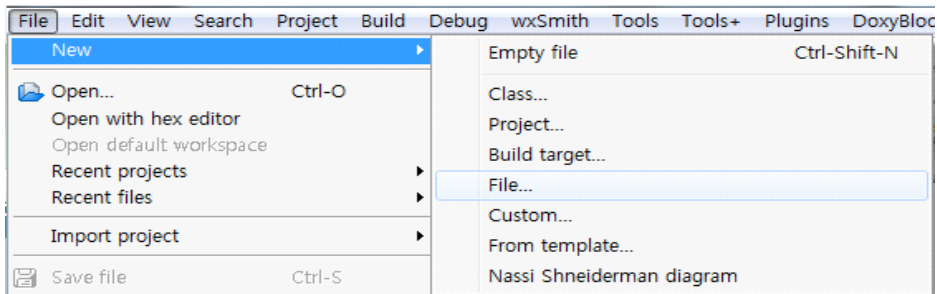


그림. 자동으로 생성된 main.cpp

⑧ File>New>File..를 선택합니다.



⑨ C/C++ source 파일을 선택합니다.

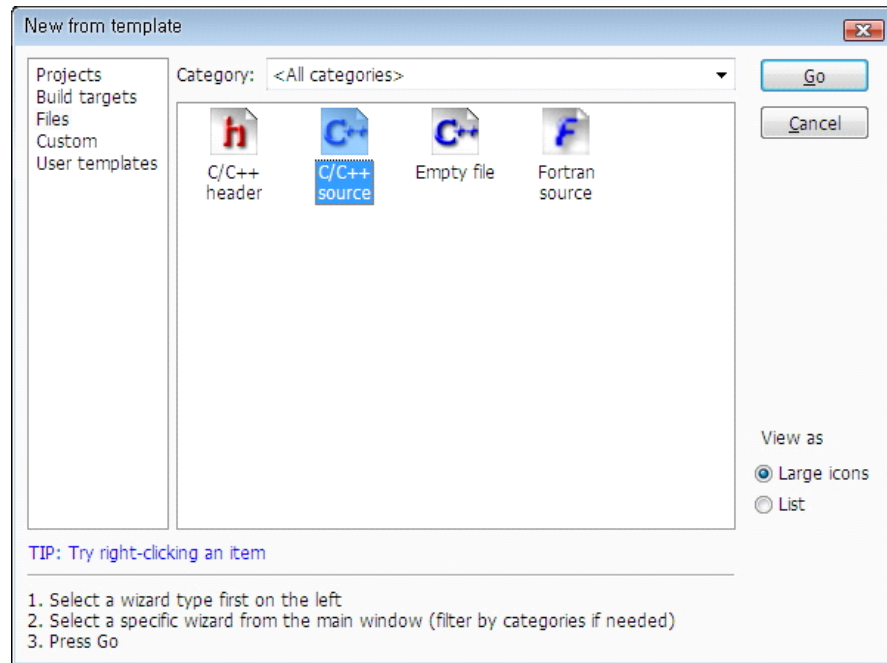


그림. 소스파일 선택

⑩ Next를 몇 번 선택하면 아래의 그림처럼 Filename을 정하고 Add file to active project를 선택해서 해당 프로젝트와 파일을 연결하고 target은 Debug와 Release 둘 다 선택한다.

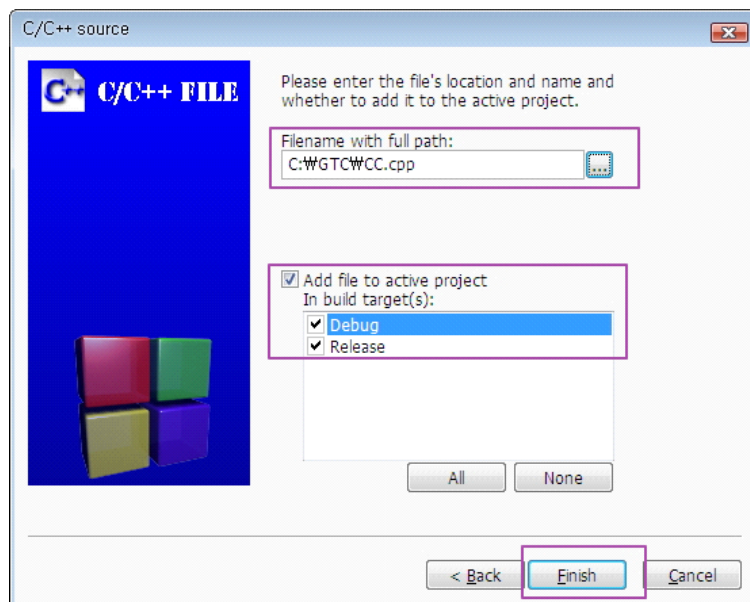


그림. 해당 파일을 활성화된 프로젝트와 연결

⑪ 아래 그림과 같이 프로젝트에 최종적으로 파일이 추가된 것을 볼 수 있다. 이제 아래의 파일에 필요한 내용을 작성한다.

※ 빠르게 파일을 추가 하는 방법은 Ctrl + Shift + n을 입력하면 된다.

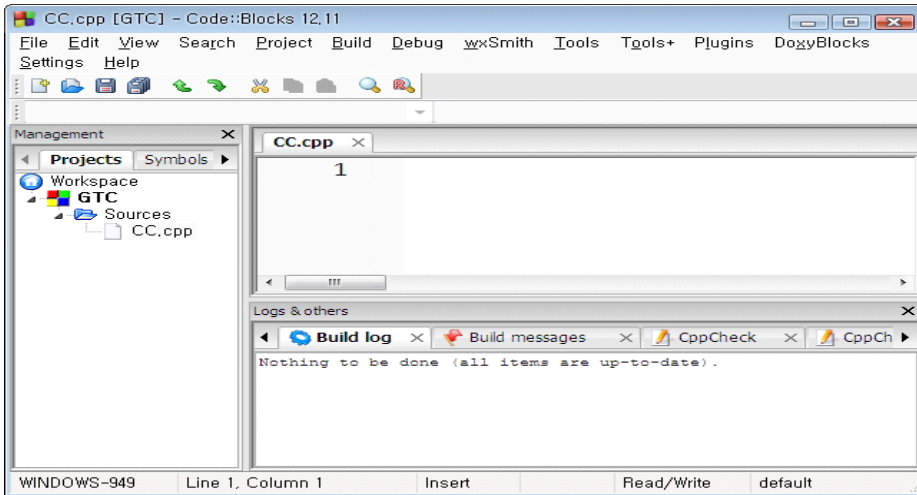


그림. 생성된 cpp파일

⑫ 다 작성된 프로그램을 실행하는 방법은 아래와 같이 Build메뉴에서 Build and run를 실행하면 된다. 원래는 Compile current file을 하고 Build하고 Run를 해야 한다. 그러나 Build and Run를 하면 빠르게 실행해서 결과를 볼 수 있다.

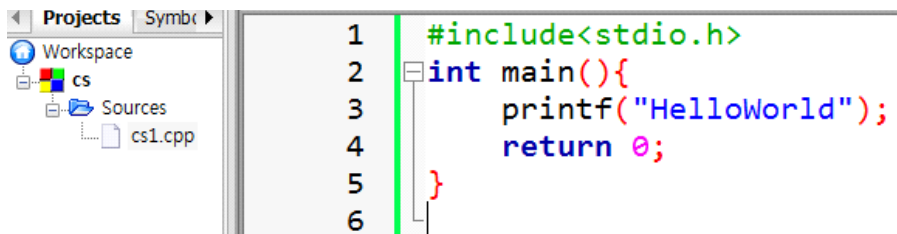


그림. 코드작성하기

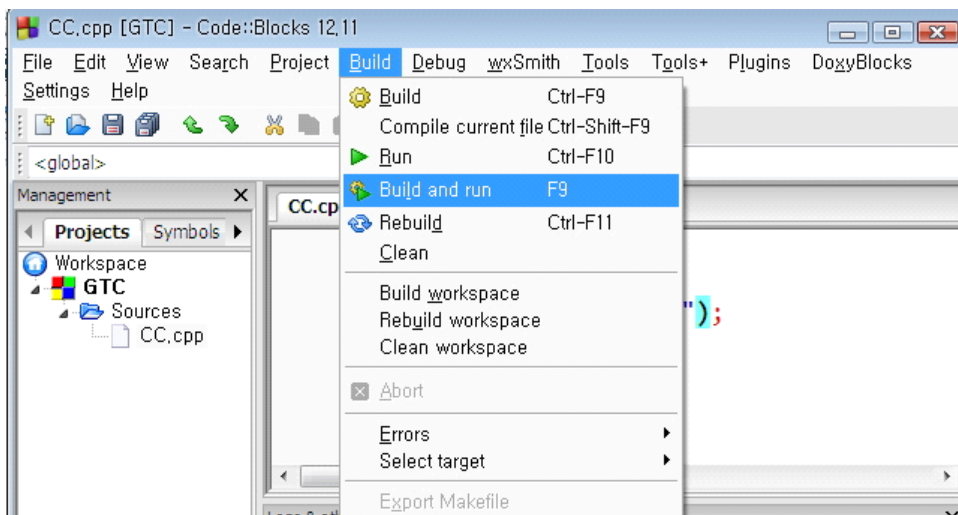


그림. 코드 작성 후 컴파일과 실행하기

2. 프로그래밍언어의 기초



대부분의 프로그래밍 언어들은 내용적인 측면에서 상당한 차이를 보이고 있다. 아주 많은 언어가 존재하지만 본교에서는 구조적 프로그래밍 언어의 대표적인 C 언어를 가지고 문제해결과 알고리즘을 기술할 것이다.

가. 문제해결 방법의 표현 (p-249)

1) 용어

알고리즘(algorithm)이란? (아라비아 수학자 알콰리즈미의 이름에서 유래)

‘어떤 목표를 달성하기 위해 실제로 수행될 수 있는 구체적인 명령의 유사한 순서와 실행절차’를 말한다.

알고리즘의 설계하는 기본적인 방법은 순차, 선택, 반복의 구조를 사용하는 것이다.

2) 알고리즘을 표현하는 방법

- 가. 자연어로 표현하기
- 나. 순서도로 표현하기
- 다. 의사코드로 표현하기
- 라. 프로그래밍 언어로 표현하기

3) 예제

1부터 100까지의 합계를 계산하는 문제를 해결하고자 한다. 어떻게 해야 할까요?

가. 자연어로 표현하기

1. 1부터 100까지 1씩 증가하면서 값을 누적해서 더한다.
2. 더하고자 하는 수 가 101이 되면 끝낸다.
3. 누적한 합을 표시한다.

나. 순서도로 표현하기

```

graph TD
    Start([시작]) --> Init[A = 0  
B = 1]
    Init --> LoopStart(( ))
    LoopStart --> Add[A = A+B]
    Add --> Inc[B = B+1]
    Inc --> LoopStart
    LoopStart --> Cond{B > 100}
    Cond -- No --> Inc
    Cond -- Yes --> Print[/A/]
    Print --> End([끝])
    
```

기호	기호의 설명	보기
	순서도의 시작이나 끝을 나타내는 기호	
	값을 계산하거나 대입 등을 나타내는 처리 기호	
	조건이 참이면 '예', 거짓이면 '아니오'로 가는 판단 기호	
	서류로 인쇄할 것을 나타내는 인쇄 기호	
	일반적인 입·출력을 나타내는 입·출력 기호	
	기호를 연결하여 처리의 흐름을 나타내는 흐름선	

다. 의사코드(pseudo- 유사한)

1. $A \leftarrow 0$
2. $B \leftarrow 1$
3. $B \leq 101$ 보다 작을 동안 반복
4. $A \leftarrow A+B$
5. $B \leftarrow B+1$
6. print A

라. 프로그래밍 언어로 표현하기

```

C언어로 표현하기
1. #include<stdio.h>
2. int main(){
3.     int A=0,B=1;
4.     while(B<=100){
5.         A=A+B;
6.         B=B+1;
7.     }
8.     printf("%d",A);
9. }
    
```

4) 알고리즘의 분석과 비교

어떤 문제를 해결할 수 있는 알고리즘은 다양하다. 실행결과만 보면 결과는 같지만 필요한 시간이나 기타 조건이 다를 수 있다. 따라서 알고리즘을 분석하는 방법을 알고 있어야 한다.

가) 예제1

똑같은 모양의 카드가 있다. 우리는 카드를 볼 수 없다. 대신 두 카드의 크기를 비교하는 로봇이 있다. 카드들을 오름차순으로 줄을 세우려고 한다. 어떻게 해야 할까?

- ① 비교하는 로봇을 몇 번 사용했는가?
- ② 최소한 로봇을 사용하려면 어떻게 해야 할까?
- ③ 더 빨리 하려면 어떻게 해야 할까?

나) 예제2

정렬되어 있는 1~100까지 임의의 수가 있다. 우리가 찾고자 하는 숫자가 이 안에 있다고 하면 과연 평균 몇 번이면 찾을 수 있을까요?

※ Big-O('빅-오 표기법)

수행시간을 기준으로 알고리즘의 효율 비교에 많이 사용하는 방법이다.

예시) $O(1)$, $O(\log_2 n)$, $O(n)$, $O(n \log_2 n)$, $O(n^2)$

나. 프로그래밍 언어 시작하기 (p-261)

0) 기초상식

프로그래밍 : 컴퓨터가 실행해야 할 명령을 작성하는 것

프로그래밍 언어 : 프로그래밍할 때 사용하는 언어

ex) 기계어, 어셈블리어, C, C++, Python, Java ...

1) 문제 상황

별다른 입력은 없고 Welcome를 출력하기. 단, 대소문자에 유의하시오

입력

없음

출력

Welcome

2) 출력화면



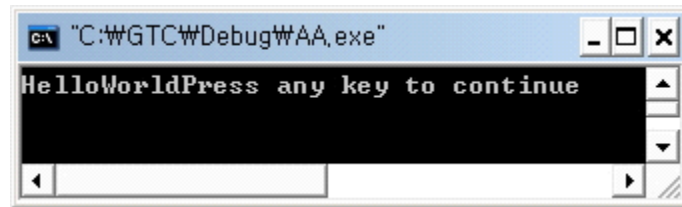
3) 소스코드

```
1. #include <stdio.h>
2. int main()
3. {
4.     printf("Welcome");
5.     return 0;
6. }
```

4) 미리 알아야 할 것들

가) 먼저 알고 있어야 하는 내용을 보면

1. C언어에서 코드는 영문자인 경우 대소문자를 구분한다. 출력결과에도 대소문자가 다르면 다른 것으로 인식한다.
2. C언어는 절차적 언어이다. 이는 명령어의 순서대로 실행된다고 보면 된다. 그래서 교재 내용에 실행코드에서 번호를 일부러 표시해 두었다. 그러나 반드시 번호대로 수행되지는 않는다. 프로그램 실행 흐름상 변경될 수도 있다.
3. 프로그램의 시작은 main()부터 하며, 반드시 main()는 하나만 존재해야 한다.
4. 필요한 내용은 직접 만들거나 이미 만들어진 것을 사용해야 한다. 아래 그림을 보면 검정색 바탕에 글자가 출력된다. 이러한 출력화면을 console 화면이라고 한다. 이는 이미 만들어져 있다. 이미 만들어진 것을 사용할 때에는 약간의 노력만 필요하다. 아니면 직접 프로그램을 작성하면서 된다.



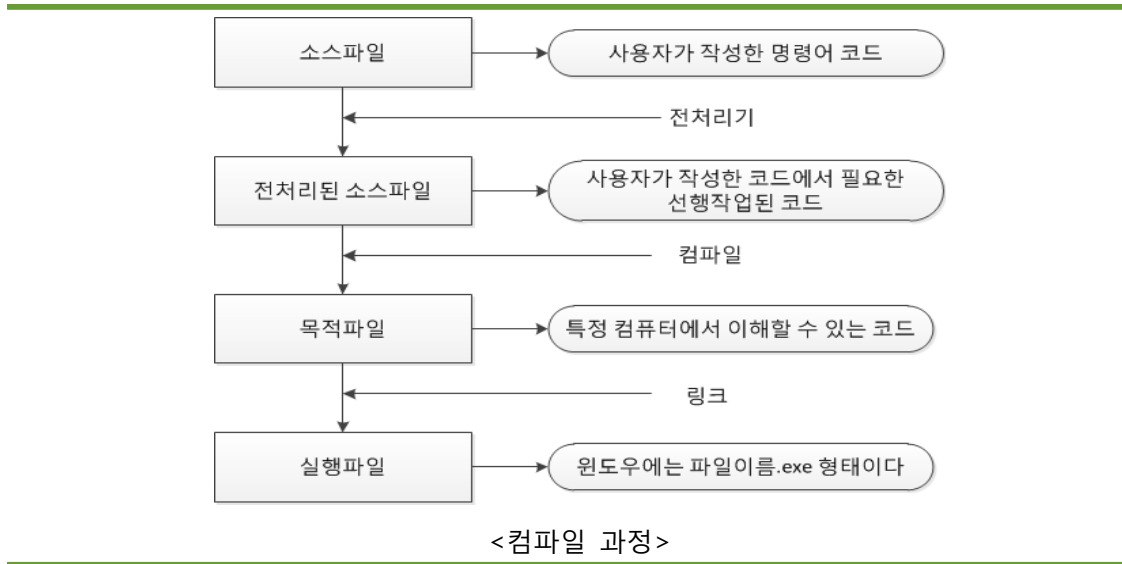
<Console 화면>

그러나 일반적으로 사용자가 console화면을 제어해서 출력하는 방법을 알아서 원하는 형태로 출력하기는 어렵다. 따라서 화면에 출력할 수 있도록 누군가 필요한 내용을 코드로 만들어 두었다. 이를 우리는 사용하면 된다. 누군가 만들어 둔 것을 사용할 때에는 아무렇게나 사용하는 것이 아니라 그에 맞는 절차가 필요하다.

이때 사용하는 명령이 #include다. 앞으로 계속 보게 될 것이다. 그리고 모든 것이 이미 만들어져 있지는 않다. 따라서 필요에 따라서는 사용자가 직접 만들어야 한다.

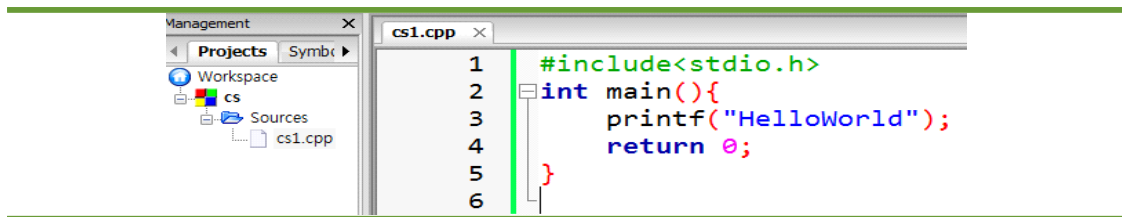
5. 블록을 기반으로 실행범위를 인식한다. 블록은 '{', '}' 로 된 형태이다.

나) 소스파일(직접 작성한)이 실행파일이 되는 과정을 보면 다음과 같다.

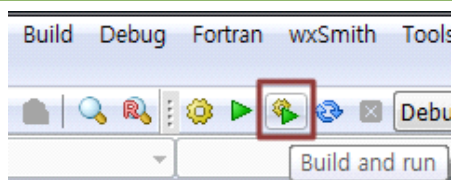


다) 실제 프로그램을 소스파일 부터 실행파일을 만드는 과정을 살펴보자.

① 사용자가 직접 코드를 작성한다.



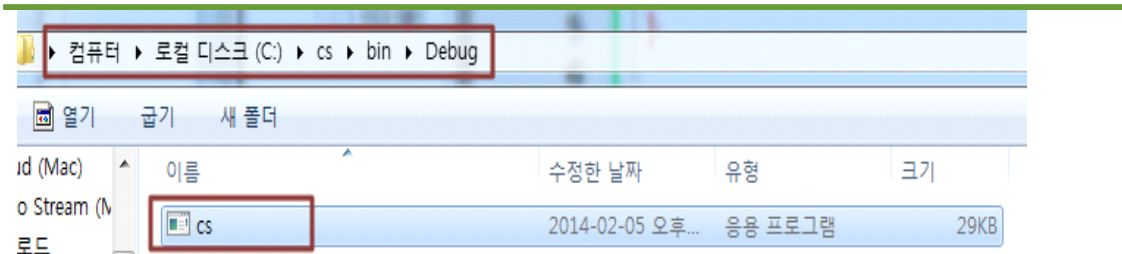
② 프로그램 컴파일 및 실행



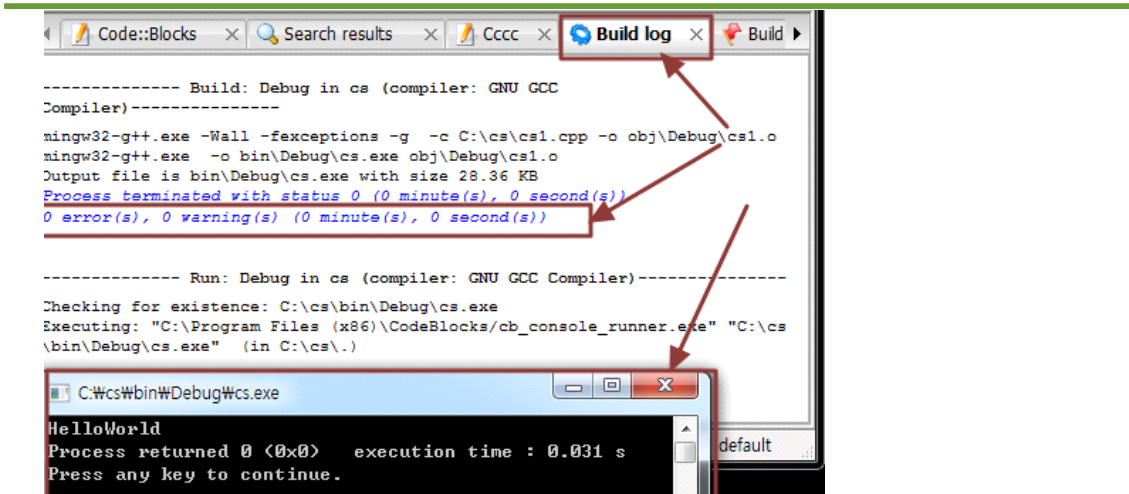
작성한 코드를 실행하기 위해 도구메뉴에서 Build and run을 실행합니다.

아니면 메뉴 Bulid에서 선택하거나 단축키 F9를 선택해도 됩니다.

③ 실행파일(.exe)



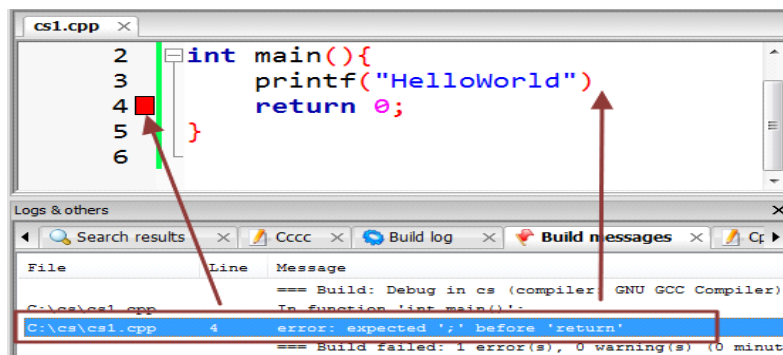
실제 윈도우상에서 작동하기 때문에 프로그램의 확장자는 .exe가 된다. 실제 위치는 프로젝트를 생성한 폴더 안에 보면 bin>Debug폴더 밑에 보면 실행파일이 있다.



Build and run를 실행하게 되면 코드에 이상이 없을 경우 틀 아래에 Build log 창에 0 error(s), 0 warning(s) 메시지를 볼 수 있고 Console 화면에 프로그램의 실행결과인 HelloWorld를 볼 수 있다. console 창을 종료하는 방법은 아무키나 누르면 종료하게 된다. 사용자 만든 소스파일의 확장자는 .cpp파일이다. 컴파일 과정을 거치면 .o(목적파일)가 만들어 진다.

만약에 코드에 이상이 있는 경우는 아래의 그림과 같은 에러메시지나 경고메시지를 출력하게 된다.

error메시지가 나오게 되면 소스코드에 문법적인 오류가 있다는 뜻이다. 아래의 그림처럼 에러난 부분 그림에서 붉은 사각형을 더블 클릭하면 소스코드 화면상에 표시된 것을 볼 수 있다. 해당 위치에 문제가 있다는 뜻이다. 물론 해당위치가 정확히 맞지 않을 수 있다. 문법적인 오류는 처음에는 이해가 되지 않아 어려운 점이 있다. 하지만 많은 오류를 경험하면 차츰 쉽게 오류를 찾아 고칠 수 있을 것이다. 자주 발생하는 오류는 철자오류, ;(세미콜론)이 없어서 발생하는 경우이다.



현재 위 코드에서는 3번 줄에 ;(세미콜론)이 없어서 발생한다. 영어를 조금 해석해 보면 이해가 될 것이다. “return 전에 ;가 기대된다.”

5) 설명

출력화면의 결과를 보면 ‘Welcome’을 출력함을 알 수 있습니다. 출력하는 부분은 4행이다. 그렇다면 나머지는 무엇인가? 처음 C언어를 공부할 때 어려운 점은 첫 시작부터 너무 많은 내용이 나온다는 것이다. ‘printf’ 만 있으면 될 것 같지만 언어이기 때문에 기본적인 문법을 지켜주어야 한다.

그러면 앞의 소스코드를 보고 분석해 보자. 먼저 실행되는 것은 2행에 main()이다. C언어에서 **영문자()로 된 형태**는 대부분 함수라고 보면 된다. 함수는 특정한 일을 수행하기 위해 일련의 작업들을 하나의 형태로 만들어 둔 것이라 보면 된다. 그 중에 **main()함수는 반드시 프로그램에 존재**해야 하며 이를 프로그램의 시작위치로 인식하기 때문이다.

3행에 ‘{’는 실행해야 할 명령들의 범위이다. 4행에 printf()가 나온다. printf()도 역시 함수이다. 또한 사용자가 만든 것이 아니기 때문에 어떻게 작동하는지 정의가 되어 있어야 한다. 이를 위해 stdio.h(standard input output header)가 있다. 이름에서 알 수 있듯이 표준입출력에 관련된 헤더이다. 결국 표준입출력에 관련된 많은 함수들에 대한 정의를 포함하고 있는 것이다. 따라서 printf()함수 외에 다양한 함수의 정의들이 포함되어 있다.

먼저 printf()는 화면출력을 담당하는 함수이다. printf()를 사용하기 위해 #include를 통해 stdio.h 파일을 불러와야 한다. 이를 통해 main()함수를 실행하기 전에 전처리(명령어들이 번역되기 전에 처리)된 소스코드가 완성된다. 대부분은 ‘#’으로 시작하는 형태는 전처리가 된다고 보면 된다.

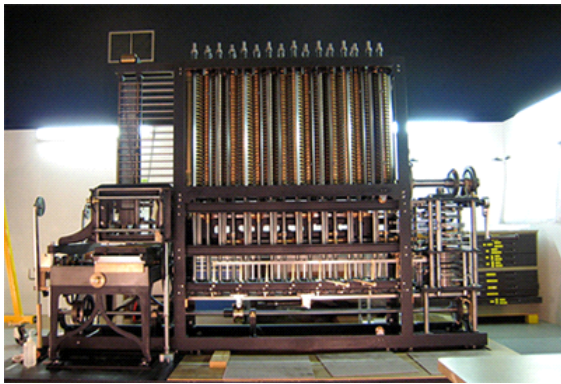
printf()의 의미는 Print formatted data to stdout이다. 이것은 표준출력에 형식화된 형태의 데이터를 출력한다는 의미이다. 무슨 말이지 잘 이해가 되지 않지만 대충 의미를 보면 “형식을 지켜서 출력해라.”

일반적으로 함수는 일련의 작업들의 묶음이기 때문에 경우 따라서는 필요한 내용을 함수에게 전달하거나 함수가 일련의 일을 수행한 후 함수의 결과 값이나 함수가 잘 실행되었는지에 대한 정보를 알려줄 필요가 있다. 2행에 int main()을 보면 함수 앞에 있는 int(integer)는 함수가 끝나고 돌려받는 반환 타입(return type)에 대한 내용이다. 여기서는 반환타입이 int(정수)가 된다. 지금은 이 정도만 기억하기 바란다.

5행에 `return 0;`의 의미는 `main()`함수가 정상적으로 잘 종료되었다는 의미로 사용되고 있다. 그러나 0대신 다른 값을 넣어도 windows환경에서는 별 문제가 없다. 그러나 `main()`함수에서는 0으로 해서 작업하는 것이 일반적이다.

※ 반환 타입에 맞게 함수 내부 코드에서 `return` 명령어를 통해 돌려받게 된다. 이때 함수의 반환 값이 있더라도 꼭 받아야 할 필요는 없다. 경우에 따라서는 함수만 실행하고 반환 값을 받을 필요가 없을 때가 있다. 반환 값이 없는 경우는 `void`를 사용하기도 한다. 보다 자세한 내용은 추후 언급된다.

프로그래밍에 수학적 사고력은 필수!



컴퓨터는 명령어로 이루어진 알고리즘에 따라 움직여요. 알고리즘을 컴퓨터가 알아들을 수 있는 언어로 쓴 것이 '코드'예요. 그런데 알고리즘은 1930년대 수학자들이 명제를 증명하기 위해 사용했던 논증방법과 비슷해요. 따라서 알고리즘을 만드는 프로그래밍을 할 때는 수학적 사고가 반드시 필요해요.

요해요.

프로그래밍을 처음 한 사람도, 세계 사람들이 가장 많이 사용하는 프로그래밍 언어인 C를 만든 사람도 수학자예요. 영국의 수학자 찰스 배비지는 삼각함수와 로그함수 문제를 빨리 계산하기 위해 1833년 '해석기관'이라고 하는 자동 계산기를 고안했어요. 천공카드에 무늬를 짜서 계산을 실행하기 위한 명령을 내렸는데, 이것이 최초의 프로그래밍이에요.

미국의 전산학자이자 수학자인 데니스 리치는 벨연구소에서 C언어를 개발했어요.

5) 실습예제

① 보이는 대로 출력하기

상황	<p>"Hello World"라는 글자를 출력해 보기. 단 띄어쓰기 및 대소문자에 유의하시오.</p> <p>입력 없음 출력 Hello World</p>
코드	<pre>1. #include<stdio.h> 2. int main(){ 3. printf(" "); 4. return 0; 5. }</pre>
CT	
의미	<p>1행: 표준입출력 함수(여기서는 printf)를 사용할 수 있게 한다.</p> <p>2행: main()함수의 시작</p> <p>3행: “ ” 안의 문자들을 그대로 출력한다.</p> <p>4행: main()함수의 반환 값 0을 return하고 프로그램은 종료된다.</p>

실전문제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=1000>

② 특수한 문자를 출력하기 (escape sequence:이스케이프시퀀스, ‘\’이용)

상황	<p>엔터키를 활용하여 화면에 글자출력하기</p> <p>입력 없음</p> <p>출력 weclome to korea</p>
코드	<pre>1. #include<stdio.h> 2. int main(){ 3. printf(" "); 4. return 0; 5. }</pre>
CT	
의미	<p>코드 상에서 엔터를 입력한다고 해서 줄 바꿈이 되지 않는다. 이러한 특수한 문자들을 출력하기 위해서 escape sequence를 이용하게 된다. 줄 바꿈(\n) 외에 존재하지만 사용할 일이 거의 없다.</p> <p>사용하는 글자체(폰트)에 따라 \n 또는 Wn 로 나오게 된다. 둘 다 같은 것이다. 이외 다른 특수한 값을 출력하고 싶으면 본 내용 뒤쪽을 참고하기 바란다.</p>

실전문제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=1001>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1003>

③ 해당 타입에 맞게 출력하기(형식필드문자이용 %)

상황	<p>주어진 정수, 문자열, 소수, 문자하나를 출력하기</p> <p>입력</p> <p>없음</p> <p>출력</p> <p>2013 now</p> <p>time : 15.220000m</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. printf("%d %s\n\ttime : %f%c",2013,"now",15.22,'m'); 4. return 0; 5. }</pre>
CT	
의미	<p>3행: “ ”에 특이하게 %문자로 된 것이 있을 것이다. 이는 “ ”안의 내용만 그대로 출력하게 될 경우 항상 동일한 결과가 나오게 된다. 그러면 여러 가지 제약사항이 많다. 그래서 형식필드(%문자)를 표시하고 각각의 의미에 맞게 “ ” 다음에 콤마(,)로 구분된 값들이 “ ” 속에 %문자에 각각에 대응되어 출력하게 된다.</p> <p>※참고사항 확인</p> <p>프로그래밍에서는 보통 문자하나와 문자열을 다르게 처리하니 유념하기 바란다.</p>

실전문제 : <http://www.judgeon.net/JudgeOnline/problem.php?id=1002>

④ 원하는 자릿값으로 출력하기

<p>상황</p>	<p>3.246을 소수 세 번째 자리에서 반올림하여 소수 두 번째 자리까지 출력하고 정수 1234는 5자리로 표시하고 빈칸은 0으로 출력하라.</p> <p>입력 없음</p> <p>출력 3.26 01234</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. printf("%.2f %05d",3.256,1234); 4. return 0; 5. }</pre>
<p>CT</p>	
<p>의미</p>	<p>3행: 형식필드에서 숫자를 포함할 경우 사용자가 원하는 형태로 출력의 변화를 줄 수 있다. 먼저 .2f는 소수 두 번째 자리까지 표시(소수 세 번째 값 반올림)이고 05d는 5자리로 표시하고 나머지는 0으로 채우라는 의미이다. 이외에도 다양한 것이 있다. 더 자세히 알고 싶은 분을 찾아보기 바란다.</p>

실전문제 : <http://www.judgeon.net/JudgeOnline/problem.php?id=1012>

※참고 사항

✦ 형식지정필드 ✦

지정자	형식	비고
%d	부호 있는 10진 정수	decimal
%u	부호 없는 10진 정수	unsigned
%lld	부호 있는 64bit 10진 정수	소문자 엘(1)을 의미함
%llu	부호 없는 64bit 10진 정수	소문자 엘(1)을 의미함
%o	부호 없는 8진 정수	octor
%x	부호 있는 16진 정수	%X는 A부터 대문자로 출력(Hexa)
%f	부호 있는 32bit 부동소수점	3/2를 출력할 경우 1.500000출력(float)
%lf	부호 있는 64bit 부동소수점	
%e	부호 있는 과학식 표현 (유효자리 와 지수승 이용)	3/2를 출력할 경우 1.500000e + 000출 력(exponential)
%g	%f, %e중 더 짧게 표현할 수 있는 표현법	3/2를 출력할 경우 1.5출력
%c	단일 문자 (아스키코드 출력)	문자 하나의 출력(character)
%s	문자열 출력	string

참조 : <http://www.cplusplus.com/reference/cstdio/printf/?kw=printf>

✦ C언어의 예외문자(Escape sequence) ✦

예외 문자	의 미	비고
\a	beep음 출력	뽁 소리가 남
\b	백스페이스	앞 글자 하나가 지워짐
\n	줄 바꿈	엔터키의 효과
\r	캐리지 리턴	그 줄의 맨 처음으로 커서 옮김
\t	탭 문자	설정된 칸만큼 띄움
\\	백슬래시 출력	특수 문자 출력
\'	작은따옴표 출력	특수 문자 출력
\"	따옴표 출력	특수 문자 출력

※ %%를 연속으로 사용하면 '%'를 출력할 수 있다.

다. 데이터 저장하기 (p-277)

1) 문제 상황

변수를 활용하여 값을 저장하고 이를 각각 출력하기

입력

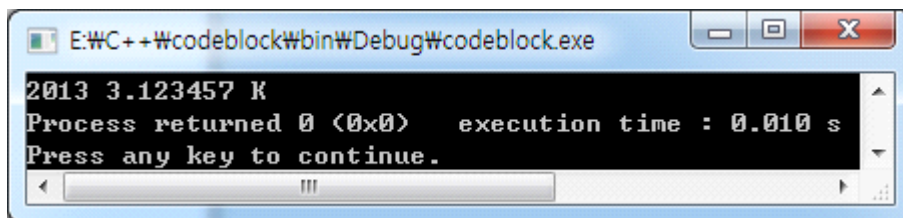
없음

출력

2013 3.123457 K

2) 출력화면

숫자나 글자를 틀에 맞게 출력한다.



3) 소스코드

```
1. #include<stdio.h>
2. int main(){
3.     int a;
4.     float b;
5.     char c;
6.     a=2013;
7.     b=3.123456789;
8.     c='K';
9.     printf("%d %f %c",a,b,c);
10.    return 0;
11. }
```


4) 설명

이번 주제는 변수를 이해하는 것이다.

변수를 사용하기 전에 C언어에서는 int a; 등의 형태로 정의를 하고 사용하게 된다.

가) 변수의 타입

변수(Variable)란 변하는 수를 말한다. ‘변하는 값을 저장하는 이름이다’라고 생각하는 것이 쉬울 것이다. C언어에서는 수를 다룰 때 변수를 정의하고 한번 정의한 후에는 같은 형태의 값만 저장하는 것이 원칙이다. 따라서 해당 변수가 저장할 수 있는 값의 형태를 정의를 해 주어야 한다. 변수가 저장할 수 있는 값의 형태가 변수의 타입(type)이 된다.

타입	의미	형식필드문자
int	32bit 크기의 부호 있는 정수 저장 ($-2^{31} \sim 2^{31}-1$ 인 정수)	%d
long long int	64bit 크기의 부호 있는 정수 저장 ($-2^{63} \sim 2^{63}-1$ 인 정수)	%lld
float	32bit 크기의 부호 있는 실수 저장 ($-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$ 인 실수)	%f
double	64bit 크기의 부호 있는 실수 저장 ($-1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$ 인 실수)	%lf
char	8bit 크기의 문자 저장 실제는 정수 값으로 처리(문자는 ASCII코드를 이용, $-2^7 \sim 2^7-1$ 인 정수)	%c(단일문자) %s(문자열)

※ 아스키 참조 : <http://www.asciitable.com/>

위에 보며 ‘부호 있는’의 뜻은 음수 값부터 양수 값까지를 의미한다. int는 $-2^{31} \sim 2^{31}-1$ 인 정수를 저장할 수 있다. 이는 2진법의 정수를 표현할 때 32bit가 가지는 값의 범위가 된다.

따라서 타입에 따라 저장할 수 있는 범위가 다르기 때문에 처리하고자 하는 값의 범위를 보고 변수의 타입을 결정해야 한다. 그리고 앞서 printf()를 다룰 때 사용한 형식필드문자도 변수의 타입이 결정된 것을 바탕으로 변수의 타입에 맞게 형식필드문자를 사용해야 한다.

※ 위의 변수 선언을 잘 보면 문제점이 있다는 것을 알 수 있다. 무엇일까요?

나) 변수의 이름

변수의 이름은 특정 타입에 맞는 값을 저장하고자 하는 영역의 이름이 된다. 따라서 정하는 규칙이 몇 가지 있다. 첫 글자는 영어대소문자 또는 ‘_’(underbar)로 시작해야 하며 그 다음 부터는 영어대소문자, 숫자, ‘_’로 구성할 수 있다.

가능한 형태로는 `_a`, `ab`, `Ab`, `A123`, `B12_` 등등으로 가능하다. 그리고 중요한 것은 대소문자가 다르면 다른 것으로 인식한다. 그리고 변수의 이름을 사용할 수 없는 것이 있다. 보통 파란색으로 표시되는 예약어들이다. 예약어에는 `auto`, `break`, `case`, `cdecl`, `const`, `char`, `continue`, `default`, `do`, `double`, `else`, `enum`, `extern`, `float`, `for`, `goto`, `if`, `int`, `long`, `register`, `return`, `short`, `signed`, `sizeof`, `static`, `void`, `volatile`, `while` 등은 사용할 수 없다.

예약어 외는 사용할 수 있지만 바람직하지 않는 변수명이 있다. 자주 사용하는 표준입출력함수나 문자열 함수, 수학함수 등이 이에 해당된다. `printf`, `scanf` 등등이 이미 만들어진 함수들의 이름이다. 이를 변수 명으로 사용할 수 있지만 절대 추천하지 않는다.

다) 변수가 사용하는 값

흔히 변수가 가지는 값 자체를 말하는 것으로 **상수**라고 할 수 있다. 변하지 않는 수를 이야기 한다. 일반적으로 숫자 자체만을 생각하기 쉬우나 프로그래밍 언어에서는 상수를 정의해서 사용할 수도 있고, 문자도 역시 상수로 처리된다.

10진 상수 : 15, 2013

16진 상수 : 0x34, 0x13FC (숫자 0과 문자 x가 앞에 포함된 형태)

부동소수점 상수 : 3.14, 5.312 (흔히 이야기하는 실수)

문자상수 : ‘A’, ‘c’ (작은따옴표 묶여 있는 형태)

문자열상수 : “GNE” (큰 따옴표로 묶여 있는 형태)

기호상수 : `#define PI 3.14` (3.14대신 PI를 사용),

`const float G=0.98` (변수 G를 상수로 취급하는 형태로 변수이지만 변수 값을 변경하는 것을 불가능하도록 한다)

라) 변수의 초기화

변수를 사용할 때에는 변수 선언이 있고 값을 할당하는 것이 있다.

선언 : `int a;` --> 변수 타입과 이름에 대한 정의를 하는 것이다.

할당 : `a=10;` --> 변수에 해당되는 값을 할당하는 것이다.

`int a=10;` --> 선언과 할당을 한 번에 할 수도 있다.

그러나 변수는 선언만 하고 참조를 할 경우 변수에는 보통 쓰레기 값(아무 의미 없는 값이 자동으로)이 할당된다. 그래서 값이 할당되지 않는 변수를 사용하면 원하는 결과를 초래할 수 있다.

물론 자동으로 초기화가 되는 경우도 있다. 특정 함수나 블록에 포함되지 않는 형태의 변수 선언이 이에 해당된다. 이를 보통 전역변수라고 한다. 전역변수는 선언과 동시에 자동으로 0으로 설정된다. 자세한 사항은 추후에 설명한다.

프로그램이 정해 준 심야버스 노선!



2013년 서울시는 0시부터 5시까지 운행하는 심야버스를 선보였어요.

심야시간 통화량과 심야택시 승하차 데이터를 활용하면 가장 적절한 심야버스

노선을 개발, 이처럼 수많은 데이터로부터 의미 있는 정보를 찾는 것을 빅데이터 분석이라고 해요. 필요한 데이터를 구별하는 안목과 선별한 데이터를 활용할 수 있는 알고리즘을 설계하는 능력이 필요하죠. 이때 프로그래밍 실력이 빛을 발한답니다.

5) 실습예제

① 원의 면적과 속도 변화 출력해 보기

<p>상황</p>	<p>속도가 3.15 m/s인 물체가 1초 동안 이동한 거리 출력하고 반지름이 3인 원의 면적을 출력하라. 단 π는 3.14로 한다.</p> <p>입력 없음</p> <p>출력 31.500000 28.260000</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. #define PI 3.14 3. int main(){ 4. const float t=1; 5. float v=31.5; 6. float r=3.0; 7. printf("%f %f",v*t, PI*r*r); 8. return 0; 9. }</pre>
<p>CT</p>	
<p>의미</p>	<p>2행: 기호상수 PI선언 4행: 시간변수 a를 상수화 시킨다. 5행, 6행: 각각 속도와 반지름을 입력한다. 7행: 이동거리와 원의 면적을 출력한다.</p>

② 쓰레기 값 출력하기

<p>상황</p>	<p>정수 변수를 선언하고 선언된 변수에 입력된 값을 출력해 보자</p> <p>입력</p> <p>없음</p> <p>출력</p> <p>-858993460</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a; 4. printf("%d",a); 5. return 0; 6. }</pre>
<p>CT</p>	
<p>의미</p>	<p>3행: 변수 선언만 하고 값을 할당하는 부분이 없다.</p> <p>4행: 출력에서 아무 의미 없는 쓰레기 값이 출력된다.</p> <p>이때 출력되는 쓰레기 값은 컴퓨터에 따라 달라질 수 있다.</p> <p>컴퓨터에 따라 달라질 수 있는지에 대해 알고 싶은 분들은 직접 컴퓨터 메모리 관리 부분에 대해 찾아보길 바란다.</p>

라. 입력받고 출력하기 (p-277)

1) 문제 상황

키보드로부터 정수 a, b와 실수 c, d를 입력받고 첫 줄에 정수 값을 출력하고 둘째 줄에 실수를 출력하라, 단, $1 \leq a, b \leq 10,000$ 인 정수이고, $0.0 \leq c, d \leq 10,000.0$ 인 실수이다.

입력

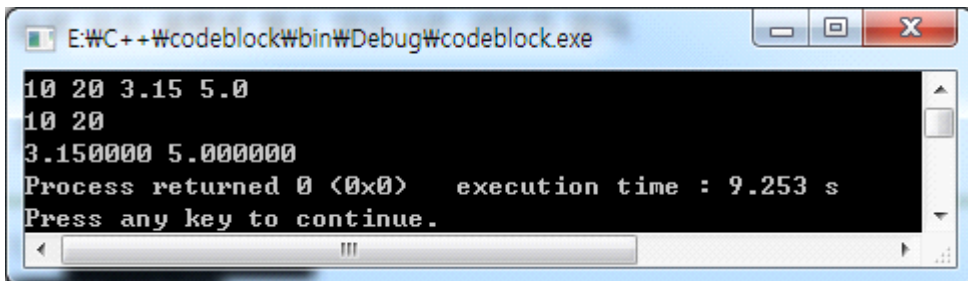
10 20 3.15 5.0

출력

10 20

3.15 5.0

2) 출력화면



3) 소스코드

```

1. #include<stdio.h>
2. int main(){
3.     int a,b;
4.     float c,d;
5.     scanf("%d%d%f%f",&a,&b,&c,&d);
6.     printf("%d %d\n%f %f",a,b,c,d);
7.     return 0;
8. }
```

4) 설명

이번 내용의 중요한 것은 입력을 받는 함수인 scanf()이다. 이 함수 역시 사용자가 만든 함수가 아니다. 따라서 이 함수의 정의는 역시 stdio.h에 포함되어 있다. scanf()함수의 의미는 Read formatted data from stdin이다. 표준입력으로 형식화된 데이터를 읽는다는 것이다. 따라서 printf()와 유사한 형식에 맞게 작성하면 된다.

여기서 중요한 것은 입력 받을 때와 출력 할 때 함수가 다르다는 것이다.

3행~4행 변수를 선언,

5행 표준입력(키보드)으로 4개의 값을 입력 받는다. 이때 형식필드를 printf()와 동일한 형태를 사용한다. 정수를 위한 %d, 실수를 위한 %f에 대응하여 입력하면 “ ” 다음에 a, b, c, d에 차례대로 대응되어 입력되어 진다.

여기서 조금 특별한 것이 있다. &(ampersand)는 scanf()에서 아주 중요한 것으로 처음 C언어를 접하는 사람은 많이 틀리는 부분이다. 변수도 결국 컴퓨터 어디인가에 저장이 된다. 변수의 실제적인 저장위치를 scanf는 요구하기 때문에 반드시 &를 변수 앞에 붙여 변수의 물리적 주소가 오도록 해야 한다.

scanf()에는 반드시 변수의 주소를 적어 주어야 한다. 아직 어렵기 때문에 일단 scanf()를 통해 입력받을 때 변수 앞에 &를 사용해야 한다는 정도만 기억하기 바란다.

6행 입력 받는 정수와 실수를 각각 printf()를 통해 출력하게 된다.

2013 노벨 화학상 수상자도 프로그래머?

2013년 노벨 화학상은 복잡하고 큰 분자의 화학반응을 컴퓨터에서 계산하고 연구할 수 있는 시뮬레이션 프로그램 '참(CHARMM)'을 개발한 과학자들이 받았어요. 참은 직접 실험을 하지 않고도 고분자의 움직임과 화학반응을 예측할 수 있는 컴퓨터 프로그램이에요. 화학은 물론 생명과학, 소재공학, 신약 개발에 까지 쓰이고 있어요.

원래 아리에 와르셀 교수는 효소를 연구하고 있었어요. 그런데 기존 시뮬레이션 프로그램으로는 도저히 이 연구를 진행할 수가 없었어요. 그래서 새로운 프로그램을 개발하는 데 착수했고, 그게 노벨상 수상으로 이어졌지요. 즉 프로그래밍을 통해 화학 연구를 한 단계 발전시킨 거예요.

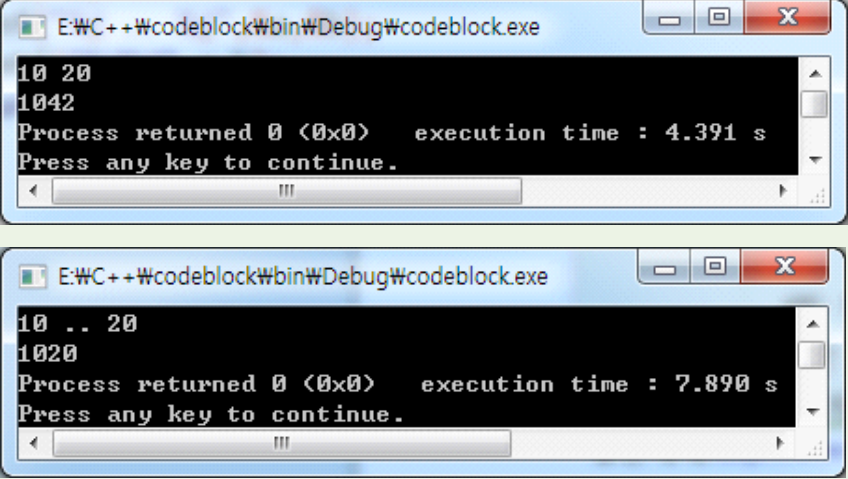
5) 실습예제

① 의미 없는 띄어쓰기

상황	<p>입력으로 두 정수 a, b가 입력된다. 입력 받은 두 정수를 그대로 출력하라. 단, $-1,000 \leq a, b \leq 1,000$인 정수이다.</p> <p>입력</p> <p style="padding-left: 40px;">10 20</p> <p>출력</p> <p style="padding-left: 40px;">1020</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",&a,&b); 5. printf("%d%d",a,b); 6. return 0; 7. }</pre>
CT	
의미	<p>3행: 두 정수를 저장하기 위해 a, b변수를 선언.</p> <p>4행: 키보드로부터 입력받기 위해 scanf함수를 사용.</p> <p>5행: 입력받은 두 정수 a,b를 printf를 통해 출력.</p> <p>화면을 보면 scanf()함수는 공백 또는 enter를 구분인자(입력을 받아들이는 단위)로 판단하기 때문에 입력받는 수의 사이에 띄어쓰기와 enter를 하나의 값을 구분하는 의미로 해석하기 때문에 띄어쓰기나 enter가 여러 번 나와도 하나로 처리하게 된다.</p>

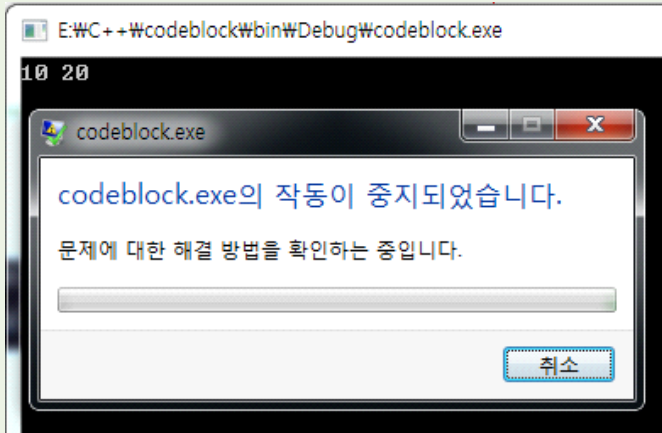
실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1010>

② 정해진 모양으로 입력받기

<p>상황</p>	<p>입력으로 정수(a)와 두 개의 (.)과 정수(b) 순으로 입력된다. 이때 입력받은 두 정수만 출력하라. 단 $-1,000 \leq a, b \leq 1,000$인 정수이다.</p> <p>입력 10 .. 20</p> <p>출력 1020</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d .. %d",&a,&b); 5. printf("%d%d",a,b); 6. return 0; 7. }</pre>
<p>CT</p>	
<p>의미</p>	 <p>4행: %d .. %d 로 형식필드가 정의된 경우 scanf()에서는 반드시 정해진 모양대로 ‘정수 .. 정수’ 형태로 입력해야 된다. 이를 어길 경우 예제1처럼 원하지 않는 결과를 볼 수 있다.</p>

실전예제:<http://www.judgeon.net/JudgeOnline/problem.php?id=1014>

③ ‘&’의 중요성 알아보기

<p>상황</p>	<p>입력으로 두 정수 a, b가 입력된다. 입력 받은 두 정수를 그대로 출력하라. 단 $-1,000 \leq a, b \leq 1,000$인 정수이다.</p> <p>입력 10 20</p> <p>출력 1020</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",a,&b); 5. printf("%d%d",a,b); 6. return 0; 7. }</pre>
<p>CT</p>	
<p>의미</p>	 <p>4행: scanf()에서 변수에 값을 입력받을 경우 변수 앞에 ‘&’가 없어 나타나는 오류이다. 경우에 따라서는 오류가 나오지 않을 수도 있다. 그렇다고 정상적으로 입력을 받은 것은 아니니 오해가 없길 바란다.</p>

마. 파일로 입출력하기

1) 문제 상황

두 정수 20, 30을 output.txt파일에 공백으로 구분하여 출력하기

입력

없음

출력

20 30

2) 출력화면

20과 30 두 정수를 output.txt파일로 출력하기 때문에 콘솔화면에는 나오지 않는다.



3) 소스코드

```
1. #include<stdio.h>
2. int main(){
3.     freopen("output.txt","w",stdout);
4.     int a=20, b;
5.     b=30;
6.     printf("%d %d",a,b);
7.     fclose(stdout);
8.     return 0;
9. }
```

4) 설명

소스코드에서 3행과 7행에 있는 내용만 처음 보는 것이다. freopen()과 fclose()이라는 함수를 사용한 것이다. 이외에도 파일로 입력과 출력을 위해 다양한 함수가 사용된다.

freopen()는 Reopen stream with different file or mode이다. 다른 파일 또는 다른 모드로 다시 스트림을 연다는 것이다. 스트림은 일련의 데이터 요소라고 생각하면 된다.

freopen(파일명,모드, 파일스트림)에서 괄호에 안에 입력되어야 할 것을 보면 먼저 입력 또는 출력하고자 하는 파일의 이름이다. 다음으로 파일을 입력 위한 것인지 출력을 위한 것인지를 나타내는 모드이다. 마지막으로 데이터의 흐름을 나타내는 파일스트림이 온다. 3행의 의미는 표준출력(stdout)스트림을 output.txt파일에 써라(write)이다.

input.txt파일로 입력 받기 위해서는 freopen("input.txt","r",stdin);로 작성하면 된다. 여기서 모드에는 w(write쓰기), r(read읽기), a(append추가) 등이 있다. 스트림에는 stdin(표준입력), stdout(표준출력)이 있다.

※참고 : <http://www.cplusplus.com/reference/cstdio/freopen/?kw=freopen>

7행의 fclose()는 사용한 파일을 닫으라는 의미이다.

결국 소스코드를 실행하면 화면상에 아무것도 출력되지 않고 대신 output.txt파일이 만들어 진다.

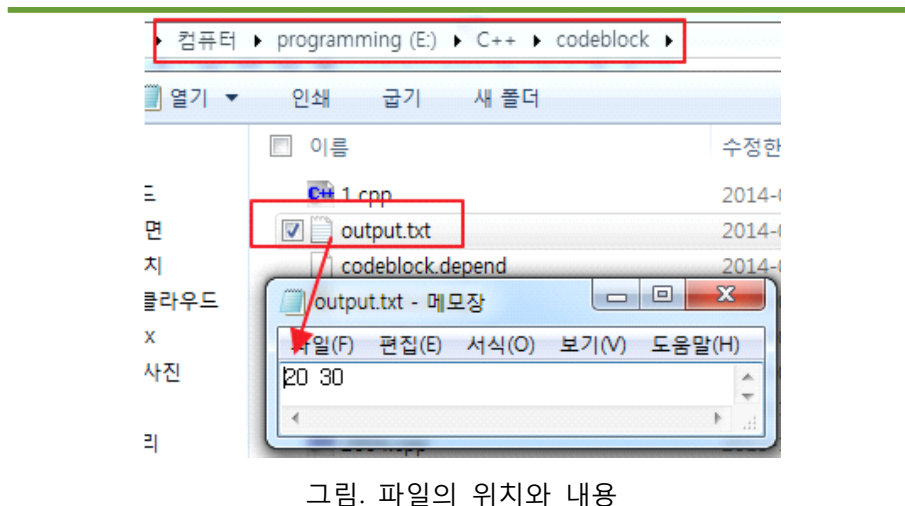
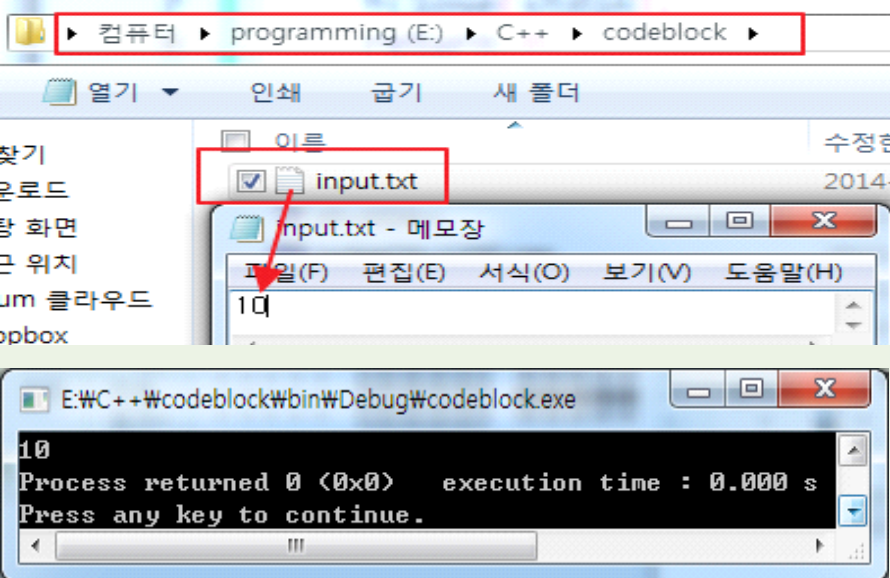


그림. 파일의 위치와 내용

간단히 생각하면 freopen()이 있느냐에 따라 파일로 입력과 출력할 것인지, 콘솔 화면에 입출력할 것인지 결정할 수 있다. 따라서 평소에는 freopen()함수 없이 코드를 작성하고 테스트한 후 마지막에 코드를 추가해서 사용하면 된다.

5) 실습예제

① 파일로 정수 입력받고 출력은 화면으로 하기

<p>상황</p>	<p>input.txt파일에서 한 정수(a)를 입력받고 입력받은 정수를 console화면에 출력하라, 단 $-1,000 \leq a \leq 1,000$인 정수이다.</p> <p>입력 파일에서 입력</p> <p>출력 10</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. freopen("input.txt","r",stdin); 4. int a; 5. scanf("%d",&a); 6. printf("%d",a); 7. fclose(stdin); 8. return 0; 9. }</pre>
<p>CT</p>	<p>만약에 input.txt파일이 없다면???</p>
<p>의미</p>	

	<p>파일로 입력받을 경우 반드시 미리 만들어져 있어야 한다. 그러나 출력파일은 프로그램이 실행되면서 생성되기 때문에 만들 필요가 없게 된다. 만약에 만들어져 있으면 w(쓰기)모드 이기 때문에 덮어쓰기가 된다.</p> <p>그림 같이 10의 정수 값을 파일로 입력받아 콘솔화면에 정상적으로 출력하게 된다.</p> <p>3행: input.txt파일을 읽기모드로 열어 표준입력으로 처리</p> <p>5행: a변수에 저장</p> <p>6행: a변수 출력</p> <p>7행: 열린 input.txt파일 닫음</p>
--	---

② 파일로 정수 입력받고 파일로 출력하기

상황	<p>한 정수(a)가 input.txt파일로 입력되고 입력된 정수에 +10을 해서 output.txt파일로 출력하라. 단 $-1,000 \leq a \leq 1,000$인 정수이다.</p> <p>입력</p> <p>10</p> <p>출력</p> <p>20</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. freopen("input.txt","r",stdin); 4. freopen("output.txt","w",stdout); 5. int a; 6. scanf("%d",&a); 7. printf("%d",a+10); 8. fclose(stdin); 9. fclose(stdout); 10. return 0; 11. }</pre>
CT	


의미	3,4행: 입출력 파일을 정의하고 6행: 파일에서 입력 7행: a+10을 해서 파일로 출력 8,9행: 생략해도 프로그램의 영향을 주지는 않는다. 그러나 파일을 열고 닫고는 꼭 해주는 것이 좋다.
----	---

③ input1.txt 파일에 정수가 두 개, input2.txt 파일에 실수가 두 개가 있다. 두 정수를 더한 결과와 두 실수를 더한 결과를 각각 공백으로 구분하여 console화면에 출력하라.

코드	
----	--

④ sample.txt 파일에 정수가 공백으로 구분되어 두 개가 있다. 이 두 수를 더해서 sample.txt 파일에 추가하라.

코드	
----	--

 이후 수업은 파일입출력을 생략하고 표준입출력형태로 진행한다. 이는 console화면에서 바로 확인하기 위한 용도이므로 만일 파일입출력이 필요하면 파일입출력 코드만 추가하면 된다. 그리고 프로그래밍 온라인으로 테스트 할 수 있는 online judge site¹⁾는 표준입출력을 대부분 구성되어 있다.

1) online judge site : 온라인에서 사용자가 작성한 코드를 제출하면 해당 시스템에서 컴파일 하여 프로그램을 실행 한 후 미리 준비된 데이터로 해당 소스코드를 분석하여 실행시간, 메모리사용량 등의 정보를 제공하여 사용자의 소스코드가 맞게 작성되었는지 온라인에서 알려주는 사이트를 말한다.

바. 계산에 필요한 연산자 (p-279)

1) 문제 상황

두 정수 a, b를 입력받아 곱하고($a*b$), 대소비교($a>b$), 부정(!a), 비트연산(a^b), 삼항 조건 연산자($a>2?a/2:b/2$)의 결과를 한 줄에 띄어쓰기로 구분하여 출력하기.

단 ($1 \leq a, b \leq 100$ 인 정수)

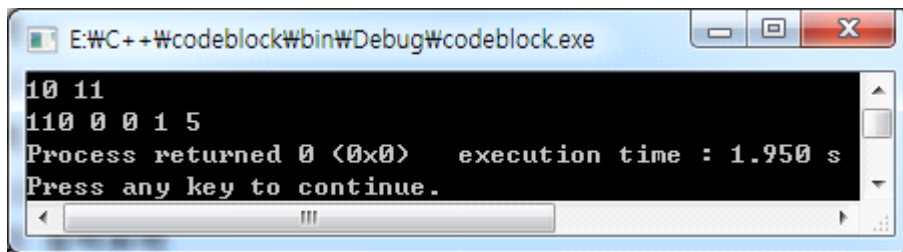
입력 :

10 11

출력

110 0 0 1 5

2) 출력화면



3) 소스코드

```

1. #include<stdio.h>
2. int main(){
3.     int a,b;
4.     scanf("%d%d",&a,&b);
5.     printf("%d ",a*b);
6.     printf("%d ",a>b);
7.     printf("%d ",!a);
8.     printf("%d ",a^b);
9.     printf("%d",a>2?a/2:b/2);
10.    return 0;
11. }
```


3) 설명

이번 내용은 연산자에 대한 내용이다. 수학에서 사용하는 기본적인 산술연산자부터 관계연산자, 논리연산자, 비트연산자, 3항 연산자에 대한 것이다. 이외에도 할당(대입)연산자, 복합대입연산자, 단항연산자 등이 있다.

예를 들어 `int a=3, b=4, c=5, d=6;`
`printf(“%d” , 예시);` 일 경우

연산자	기호	예시	출력	의미
산술	+	a+b	7	두 항을 더하라
	-	a-b	-1	두 항을 빼라
	*	a*b	12	두 항을 곱하라
	/	a/b	1	a를 b로 나누어라
	%	a%b	3	a를 b로 나눈 나머지
관계	>	a>b	0	a가 b보다 크나?
	<	a<b	1	a가 b보다 작나?
	==	a==b	0	a가 b보다 같나?
	<=	a<=b	1	a가 b보다 작거나 같나?
	>=	a>=b	0	a가 b보다 크거나 같나?
	!=	a!=b	1	a가 b랑 같지 않나?
관계 연산의 결과는 항상 참(true, 1)이거나 거짓(false,0) 이다.				
논리	&&	a>b && c<d	0	a가 b보다 크고 c가 d보다 작다
		a!=b c==d	1	a가 b랑 같지 않거나 c가 d랑 같거나
	!	!(a<b)	0	결과를 부정(a>=b 의미는 같음)
논리 연산의 결과는 항상 참(true, 1)이거나 거짓(false,0) 이다.				
비트	&	a&b	0	비트끼리 &(AND)를 한 결과
		a c	7	비트끼리 (OR)를 한 결과
	^	c^b	7	비트끼리 ^(XOR)를 한 결과
	~	~a	-4	비트를 반전(2의 보수로 표현되기 때문에 -3이 아니라 -4가 된다)
	>>	a>>1	1	비트전체를 오른쪽으로 이동(shift) (a/2와 동일)
	<<	a<<1	6	비트전체를 왼쪽으로 이동 (a*2와 동일)
삼항	조건? A:B	a>2?a/2:a*2	1	조건이 참이면 A, 아니면 B

이외에는 중요한 할당(대입)연산자(=)과 복합할당(대입)연산자(+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=), 단항증감연산자(++/--)도 있다.

할당연산자(Assignment)는 변수에 값을 부여하는 것을 말한다. 예를 들어 int a=5; 라고 할 경우 a라는 변수에 5의 값을 부여한다. 여기서 중요한 것은 오른쪽에 있는 값을 왼쪽의 공간으로(right to left)의 개념을 기억해야 한다.

복합할당연산자(Compound assignment)는 할당연산자와 다른 연산자가 결합된 형태입니다. 예를 보면 a=a+1을 a+=1로 줄여서 표시하는 형태입니다. 다른 것도 동일하다. a*=3은 a=a*3이 된다.

일반적인 연산자는 항이 두 개 필요하지만 단항이 필요한 연산자도 있다. 보통 부호를 의미하는 +와 -이다. 이 외에 증감연산자(후위, 전위)도 항이 하나만 필요한 단항이다. 그러나 증감 연산자는 단독으로 사용할 경우 a++, ++a는 a=a+1과 동일한 의미지만 할당연산자와 같이 사용할 경우 다른 결과를 가져온다.

1. int k,m; 2. k=3; 3. m=++k; k와 m는 모두 4의 값을 가진다.	1. int k,m; 2. k=3; 3. m=k++; m은 3, k는 4가된다.
전위 증가/감소 연산자는 먼저 증가/감소 되고 할당이 되지만 후위 증가/감소는 할당이 먼저 되고 나서 후에 증가/감소가 된다.	

6행은 a*b의 결과 110이고 7행은 a>b인 결과 거짓(FALSE)이므로 0이 된다. 8행은 a를 부정하니 0, 이때 C언어에서는 0을 제외한 모든 값은 참(TRUE)으로 봅니다. 따라서 10도 참(TRUE)이므로 부정하니 거짓(FALSE)가 된다. 9행은 비트끼리 XOR를 하기 때문에 a와 b는 32bit 정수형이기 때문에 2진수로 표현할 경우, a^b는

```

0000 0000 0000 0000 0000 0000 0000 1010(10) .....a
0000 0000 0000 0000 0000 0000 0000 1011(11) .....b
0000 0000 0000 0000 0000 0000 0000 0001(1) .....최종
    
```

가 되어 최종결과에서 1이 된다.

10행은 a>2보다 크기 때문에 조건이 참이므로 a/2가 실행된다. 결과는 5가 된다.

4) 실습예제

① 두 수를 곱해 보자

상황	<p>두 정수 a,b를 입력받아 a*b의 결과를 출력해 보자 단 $1 \leq a, b \leq 1,000$인 정수이다.</p> <p>입력 20 3</p> <p>출력 60</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",&a,&b); 5. printf("%d",a*b); 6. return 0; 7. }</pre>
CT	
의미	

실전예제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=1020>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1021>

② 산술연산자를 이용한 출력

<p>상황</p>	<p>두 정수 a,b를 입력받아 a를 b로 나눈 몫과 나머지를 출력하라. 단 $1 \leq a, b \leq 1,000$인 정수이다.</p> <p>입력 20 3</p> <p>출력 6 2</p>
<p>코드</p>	<pre>1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",&a,&b); 5. printf("%d %d",a/b,a%b); 6. return 0; 7. }</pre>
<p>CT</p>	
<p>의미</p>	<p>a/b 로 나누기를 하면 수학에서는 소수점이 나올 것이다. 그러나 C언어에서는 정수 / 정수를 하게 되면 정수를 출력하게 된다. 결국 소수점이 출력되지 않게 된다. 1/2는 0.5이지만 출력은 0이 된다. 이러한 결과는 형변환(typecast)에 관련된 내용이니 잘 숙지하기 바란다.</p> <p>일반적으로 정수+정수 = 정수, 정수+실수=실수, 실수+실수=실수가 된다. 따라서 여러분들이 원하는 최종 결과가 정수인지 실수인지를 결정하고 그에 맞게 정수연산을 할 건지 실수 연산을 할 건지 결정해야 한다. 형을 변환하지 않으면 원하는 결과가 나오지 않을 수 있다. 변환하는 방법은 단순히 정수 숫자를 실수로 변경할 경우 3에다 3.0으로 변경하면 된다. 그러나 변수일 경우 a에 a.0을 할 수는 없다. 따라서 강제적으로 형을 변화시켜야 한다. 이때 사용하는 것은 변환하고자 하는 타입을 적어주면 된다. (double)a 라고 표시를 하면 a를 실수형으로 형을 변환하게 된다.</p> <p>5행: a%b는 a를 b로 나눈 나머지를 출력한다.</p>

실전예제:

<http://www.judgeon.net/JudgeOnline/problem.php?id=1019>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1022>

③ 관계연산자를 이용한 출력

상황	<p>두 정수 a,b를 입력받아 a가 b보다 작으면 1을 출력하고 아니면 0을 출력하라. 단 $-1,000 \leq a, b \leq 1,000$인 정수</p> <p>입력</p> <p>3 9</p> <p>출력</p> <p>1</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",&a,&b); 5. printf("%d",a<b); 6. return 0; 7. }</pre>
CT	
의미	<p>5행: a는 3이고 b는 9이다 따라서 $a < b$는 참(true)가 되어 1을 출력하게 된다.</p>

실전예제:

<http://www.judgeon.net/JudgeOnline/problem.php?id=1026>

④ 논리연산자를 이용한 출력

<p>상황</p>	<p>두 정수 a, b를 입력받아 a가 0이 아니거나 a가 b보다 작으면 1, 아니면 0을 출력하라. 단 $-1,000 \leq a, b \leq 1,000$인 정수이다.</p> <p>입력</p> <p>3 4</p> <p>출력</p> <p>1</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",&a,&b); 5. printf("%d", a a<b); 6. return 0; 7. }</pre>
<p>CT</p>	
<p>의미</p>	<p>5행: 3과 4과 입력되기 때문에 a는 0이 아니기 때문에 참(true) 이고 a<b도 참이다. 두 조건이 모두 만족한다. 전체는 참(true)가 되어 1을 출력한다..</p>

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1027>

⑤ 비트연산자를 이용한 출력

<p>상황</p>	<p>두 정수 a와 b를 입력받아 비트끼리 and 연산결과를 출력하고 a를 b만큼 왼쪽으로 shift를 실시한 결과를 출력하라. 단 $1 \leq a, b \leq 30$ 인 정수</p> <p>입력</p> <p>2 6</p> <p>출력</p> <p>2 128</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b; 4. scanf("%d%d",&a,&b); 5. printf("%d ",a&b); 6. printf("%d",a<<b); 7. return 0; 8. }</pre>
<p>CT</p>	
<p>의미</p>	<p>5행: 2와 6이 입력되므로 2는 2진수로 010이고 6은 110이다 따라서 상위 비트는 동일하기 때문에 비트끼리 &(AND)를 하면 10이 된다.</p> <p>6행: a를 6번 왼쪽으로 이동(Shift)하면 2를 6번 곱한 것과 동일하기 때문에 27와 동일하다.</p>

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1028>

⑥ 삼항연산자를 이용한 출력

<p>상황</p>	<p>한 정수 a를 입력받아 a가 짝수이면 10, 아니면 20을 출력하라. 단 -1,000<=a<=1,000인 정수</p> <p>입력 3</p> <p>출력 20</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a, b=10, c=20; 4. scanf("%d",&a); 5. printf("%d",a%2==0? b : c); 6. return 0; 7. }</pre>
<p>CT</p>	
<p>의미</p>	<p>4행: 3이 입력</p> <p>5행: a를 2로 나누면 나머지가 1이므로 조건이 거짓이 된다. 따라서 c가 출력이 된다.</p>

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1030>

사. 코드의 흐름을 제어하자 (p-269)

1) 문제 상황

한 정수 a 를 입력받아 2의 배수이면 a 를 2로 나눈 몫을 출력하고 아니면 a 에 2를 곱한 값을 출력하라. 단 $-1,000 \leq a \leq 1,000$ 인 정수

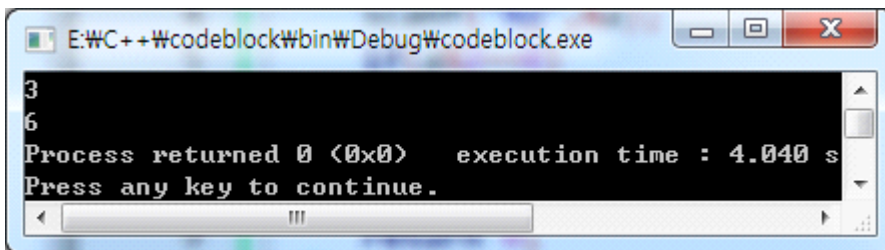
입력

3

출력

6

2) 출력화면



3) 소스코드

```

1. #include<stdio.h>
2. int main(){
3.     int a;
4.     scanf("%d",&a);
5.     if(a%2 == 0)
6.         printf("%d",a/2);
7.     else
8.         printf("%d",a*2);
9.     return 0;
10. }
```

4) 설명

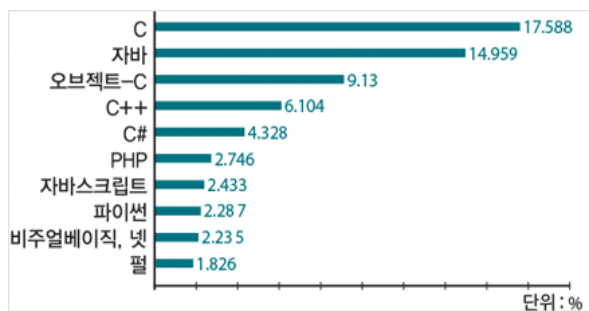
실행 흐름을 제어하는 제어문에 관련된 내용이다. 가장 많이 사용하는 제어문으로 if문을 사용한다. if문은 다양한 형태가 가능하다. 기본적인 형태만 안내하고 실습예제를 통해 익히기 바란다.

```
[기본형태]
if (조건)
    [명령...]
[else [if(조건)]
    [명령... ]
```

물론 if문 외에도 제어문에는 switch() case문이 있지만 사실 사용할 일이 그렇게 많지 않다. 필요할 경우 개별적으로 찾아보기 바란다.

5행에 if(조건)은 영어처럼 ‘만약에 ~하면’의 의미로 C언어에서도 동일한 의미로 생각하면 됩니다. 대신 조건이 참이면 실행되어야 문장과 거짓일 경우 실행될 문장으로 처리하게 된다. 5행에서 $a \% 2 == 0$ 은 a가 짝수이면 6행을 실행하고 홀수이면 8행을 실행하라는 의미이다. 결국은 6행과 8행은 둘 다 동시에 실행되지는 않는다.

세상에서 가장 많이 사용하는 프로그래밍 언어는?



매달 세계 컴퓨터 언어 사용 순위를 알려 주는 티오베 (www.TIOBE.com바로가기)의 발표에 따르면 2014년 12월을 기준으로 세계에서 가장 많이 사용하는 프로그래밍 언어는 C예요. C는 모든 컴퓨터시스템에 사용할 수 있기 때문에 세계적으로 가장 많이 사용하고 있는 언어예요. 소프트웨어 관련 직업을 희망한다면 필수로 배워야 하는 언어죠. 사실 C와 C++, 오브젝트-C, 자바, 자바스크립트는 모두 언어 구조가 비슷해요. 그래서 C언어 하나만 제대로 배우면 다른 언어를 쉽게 배울 수 있어요.

5) 실습예제

① 입력된 수가 짝수일 경우만 “even”을 출력

<p>상황</p>	<p>한 정수 a가 입력된다. a가 짝수이면 even을 출력하고 아니면 출력은 없다. 단 $-1,000 \leq a \leq 1,000$인 정수</p> <p>입력 4</p> <p>출력 even</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a; 4. scanf("%d",&a); 5. if(a%2==0) printf("even"); 6. return 0; 7. }</pre>
<p>CT</p>	
<p>의미</p>	<p>5행: if(조건) 조건이 짝수일 때에만 printf()가 실행되어 화면에 even을 출력한다. 거짓이면 6행으로 넘어가기 때문에 아무것도 출력하지 않게 된다.</p>

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1029>

② 두 정수를 입력받아 같으면 1, 다르면 0을 출력해라

상황	두 정수 a, b을 입력받아 두 정수가 같으면 1, 아니면 0을 출력하라. 단 $-1,000 \leq a \leq 1,000$ 인 정수 입력 3 9 출력 0
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a,b; 4. scanf("%d%d",&a,&b); 5. if(a==b) printf("1"); 6. else printf("0"); 7. return 0; 8. }</pre>
CT	
의미	5행: a==b가 같은 경우 1을 출력 6행: 5행의 조건이 거짓이면 0을 출력한다.

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1035>

③ 한 수를 입력받아 성적을 출력하라

<p>상황</p>	<p>한 정수 a를 입력받아 a가 90이상이면 A, 80이상이면 B, 70이상이면 C, 60이상이면 D, 아니면 F를 출력하라. 단 $0 \leq a \leq 100$인 정수</p> <p>입력</p> <p>74</p> <p>출력</p> <p>C</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a; 4. scanf("%d",&a); 5. if(a>=90) printf("A"); 6. else if(a>=80) printf("B"); 7. else if(a>=70) printf("C"); 8. else if(a>=60) printf("D"); 9. else printf("E"); 10. return 0; 11. }</pre>
<p>CT</p>	
<p>의미</p>	<p>4행: 숫자 하나를 입력받는다.</p> <p>5행-9행: 차례대로 조건을 검사한다. 90이상이면 A를 출력, 80이상이면 B를 출력, 70이상이면 C, 60이상이면 D, 나머지는 E를 출력하게 된다.</p> <p>5행부터 차례대로 비교하여 조건이 참일 경우 해당 문장만 실행하게 된다.</p>

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1030>

④ 입력된 숫자 중에 가장 작은 수 출력하기

<p>상황</p>	<p>세 정수 a,b,c가 입력된다. 세 수 중에 가장 작은 수를 출력하라. 단 -1,000<=a,b,c<=1,000인 정수 입력 3 2 1 출력 1</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a,b,c; 4. scanf("%d%d%d",&a,&b,&c); 5. if(a<b){ 6. if(a<c) printf("%d",a); 7. else printf("%d",c); 8. } 9. else { 10. if(b<c) printf("%d",b); 11. else printf("%d",c); 12. } 13. return 0; 14. }</pre>
<p>CT</p>	

의미	<p>보통 제어문을 중첩하여 사용할 수 있다. 지금 내용은 중첩된 if() 사용한 것이다. 이때 중요한 것은 하나의 if 조건이 참일 때 실행되어야 할 문장이 두 개 이상 일 때에는 반드시 ‘ { } ’가 필요하게 된다.</p> <p>4행: 3 2 1이 입력된다.</p> <p>5행: 먼저 두 수 a, b를 비교해서 a가 b보다 작은 경우 a와 c의 대소 비교를 통해 $a < c$일 경우 6행 a를 출력하고 아닐 경우 7행 c를 출력한다.</p> <p>9행: a가 b보다 크거나 같은 경우 b와 c의 대소비교를 통해 $b < c$인 경우 10행 b를 출력하고 아닐 경우 11행 c를 출력한다.</p>
----	--

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1031>

⑤ 남자인지 여자인지 출력해 보자

<p>상황</p>	<p>주민번호 뒷자리의 첫 번째 값을 나타내는 한 정수 a가 입력된다. 이를 이용해 a가 1 또는 3일 경우 "man"을 2,4일 경우 "woman"을 출력하라. 단 $1 \leq a \leq 4$인 정수</p> <p>입력</p> <p>3</p> <p>출력</p> <p>man</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a; 4. scanf("%d",&a); 5. if(a==1 a==3) 6. printf("man"); 7. else printf("woman"); 8. return 0; 9. }</pre>
<p>CT</p>	
<p>의미</p>	<p>5행: 논리연산자와 관계연산자를 한 번에 처리한 형태이다. (OR)를 사용하여 a가 1이거나 3일 경우 man을 출력하고 아닐 경우 woman을 출력한다.</p>

실전예제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1036>

아. 반복된 일을 시켜보자 (p-274)

1) 문제 상황

정수 n이 입력되면 1부터 n까지 차례대로 출력하라. 단 $1 \leq n \leq 100$ 인 정수

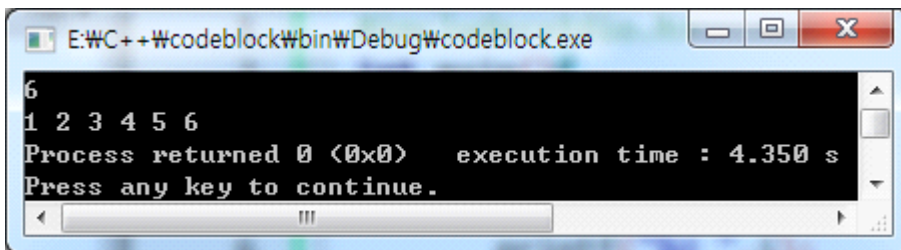
입력

6

출력

1 2 3 4 5 6

2) 출력화면



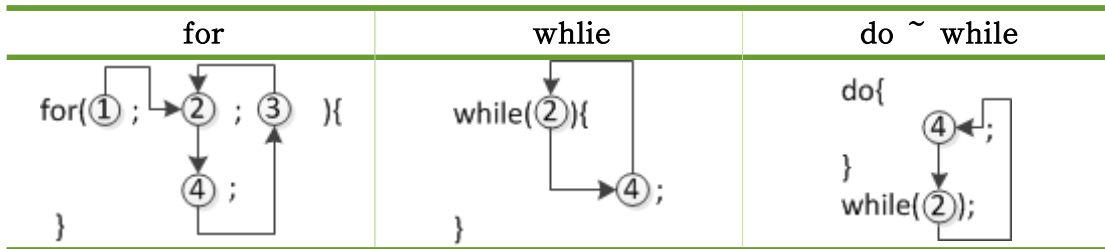
3) 소스코드

```

1. #include<stdio.h>
2. int main(){
3.     int n,i;
4.     scanf("%d",&n);
5.     for(i=1;i<=n;i++)
6.         printf("%d ",i);
7.     return 0;
8. }
```

4) 설명

프로그래밍에서 가장 기초이면서 중요한 것이 반복과 제어다. 이번의 주제는 반복된 작업을 실행하는 것이다. 크게 반복 작업에 해당하는 명령은 두 가지가 있다. for, while이다. 각각에 대해 살펴보면,



일반적인 프로그램은 무한히 반복은 없다. 따라서 반복에는 시작과 끝이 존재해야 한다.

위의 그림에서 보면 for문은 ①을 한번 실행하고 나머지는 차례대로 반복되어 실행한다. 이때 반복의 끝을 판단은 ②번에 있는 내용을 한다. 따라서 ②의 결과는 True/False로 나오게 된다. ②가 참이면 ④번과 ③을 실행하고 다시 ②를 검사하게 된다. 이때 ④는 한 문장 이상이 될 수 있다. ‘ { } ’는 여러 개 문장을 하나의 블록을 묶는 의미이다. 따라서 실행되어야 할 문장이 하나라면 ‘ { ’는 없어도 된다. 그러면 for에 있는 ①과 ③은 있어야 하는가? 실제 없어도 상관은 없다. 단지 for문이 그림과 같이 순서대로 반복되어 실행된다고 이해하기 바란다.

while은 영어에서 ~하는 동안이다. C언어에서는 비슷한 의미로 보면 된다. ②가 참일 동안 ④를 계속 실행하게 된다. 따라서 ④에서 ②의 조건에 변화를 주는 내용이 들어가야 한다. 여기서도 ④는 한 문장 이상이 될 수 있다.

do while은 일단 ④를 한번 실행하고 ②를 검사하게 된다. 나머지는 while과 동일하다.

for와 while에서 조건 ②가 없으면 어떻게 될까? 이럴 경우 반복의 종료는 없기 때문에 무한정 반복하게 된다. 또한 0을 제외한 단순한 숫자 값만 입력된 경우도 항상 참이 된다. 예를 들어 ②에 숫자로 10을 입력하면 10이라는 수는 0(false)가 아니기 때문에 항상 참이 된다. 실수하기 쉬운 부분이니 유념해서 사용하길 바란다.

소스코드를 보면 4행에서 n을 입력받는다. 예제는 6을 입력받아서 5행에서 i에 1을 할당하고 i가 6보다 작기 때문에 6행을 실행하여 1을 출력한다. i++를 실행하면 i는 2가 되고 여전히 i는 6보다 작기 때문에 6행을 실행하게 된다. 결국 i가 7이 되는 순간 i<=n 조건이 거짓이 되므로 반복이 종료되고 7행을 실행하여 프로그램은 종료된다.

5) 실습예제

① 1부터 입력된 n까지의 합을 계산하기

상황	<p>한 정수 n이 입력된다. 그러면 1부터 n까지 합을 출력하자. 단, $1 \leq n \leq 1,000$인 정수</p> <p>입력 10</p> <p>출력 55</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int n,sum=0,i; 4. scanf("%d",&n); 5. for(i=1;i<=n;i++) 6. sum=sum+i; 7. printf("%d",sum); 8. return 0; 9. }</pre>
CT	
의미	<p>3행: sum=0을 하는 이유는 초기 값이 정해지지 않을 경우 6행에 sum+i에서 sum의 값이 얼마인지 알 수가 없기 때문에 엉뚱한 계산의 결과를 가져오기 때문이다.</p> <p>5,6행: i가 1부터 n까지 반복하면서 i의 값을 누적해서 더해 sum에 넣는다.</p> <p>7행: 출력하면 1~n까지의 합을 계산할 수 있다. 물론 더 간단히 수식을 이용할 수도 있다.</p>

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1040>

② 여러 개의 입력을 받아 각각 출력하기

<p>상황</p>	<p>입력으로 개수를 나타내는 한 정수 n이 입력된다. 입력받은 n만큼의 데이터가 한 줄에 입력된다. 이때 입력된 데이터를 모두 출력하라. 단, $1 \leq n \leq 100$인 정수, $1 \leq \text{데이터} \leq 1000$인 정수</p> <p>입력</p> <pre>6 1 2 3 4 5 6</pre> <p>출력</p> <pre>1 2 3 4 5 6</pre>
<p>코드</p>	<pre>1. #include<stdio.h> 2. int main(){ 3. int n,i,temp; 4. scanf("%d",&n); 5. for(i=1;i<=n;i++){ 6. scanf("%d",&temp); 7. printf("%d ",temp); 8. } 9. return 0; 10. }</pre>
<p>CT</p>	
<p>의미</p>	<p>4행: 데이터 개수 n을 입력받는다. 현재 값은 6이다.</p> <p>5~7행: i가 n까지 6,7행을 반복하게 된다. 결국 총 6번의 6,7행을 실행하게 된다. 여기서 temp변수를 사용한 이유는 입력되는 데이터를 임시 저장하고 다시 출력하기 위해 사용되었다.</p> <p>6행은 값을 입력받아 temp라는 변수에 저장하고 7행에서 temp에 저장된 값을 출력하게 된다.</p>

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1054>

③ 2단부터 9단까지 구구단 출력하기

상황	<p>구구단을 출력하라. 단, 2단부터 9단까지로 한다.</p> <p>입력</p> <p>없음</p> <p>출력</p> <p>2*1=2</p> <p>2*2=4</p> <p>...(생략)...</p> <p>9*8=72</p> <p>9*9=81</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int i,k; 4. for(i=2;i<=9;i++) 5. for(k=1;k<=9;k++) 6. printf("%d*%d=%d\n",i,k,i*k); 7. return 0; 8. }</pre>
CT	
의미	<p>반복문 역시 중첩해서 사용이 가능하다. 어려워 보이지만 for문의 기본 순서만 잘 기억하고 흐름대로 따라가면 된다. 반복문 전체를 하나의 명령어라고 생각하여 보면 쉬울 것이다.</p> <p>4행: i는 2부터 시작하고 i는 9보다 작으므로 5행이 실행된다. 이때 k는 1부터 k가 9보다 작거나 같을 동안 6행을 실행하기 때문에 2*1=2부터 2*9=18까지 출력한다.</p> <p>5행: for문은 k가 10이 되므로 종료하게 된다. 이것은 4행에 for문 i=2일 때 반복한 것이므로 i가 3이 되고 다시 5행을 다시 실행하게 된다. 그러면 3*1=3부터 3*9=27까지 출력한다. 이러한 반복을 통해 9*9=81까지 출력하게 된다.</p>

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1059>

④ 입력된 양의 정수 5개 중 나머지가 가장 큰 것을 출력하기

<p>상황</p>	<p>두 양의 정수로 이루어진 a, b가 각각 5개씩 입력된다. 이때 a를 b로 나눈 나머지가 가장 큰 것을 출력하라. 단 $1 \leq a, b \leq 1,000$인 정수</p> <p>입력</p> <p>4 8 8 4 20 8 29 6 16 11</p> <p>출력</p> <p>5</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int i,a,b,max=0; 4. i=1; 5. while(i<=5){ 6. scanf("%d%d",&a,&b); 7. if(a%b>max) max = a%b; 8. i++; 9. } 10. printf("%d",max); 11. return 0; 12. }</pre>
<p>CT</p>	
<p>의미</p>	<p>이 문제는 while를 이용한 형태이다. for문을 이용할 것인지 while문을 이용할 것인지는 코드를 작성하는 사람이 판단해서 작성하면 된다. 둘 다 동일한 반복이 가능하기 때문이다.</p>

3행: max에 0을 넣고 시작하는 이유는 가장 큰 값을 비교할 것이기 때문에 나머지 중 가장 작은 값인 0을 미리 넣어두고 더 큰 값이 나오면 변경하기 위해서이다.

4행: i에 1로 시작하는 이유는 while문의 시작조건을 추기 위함이다.

8행: i를 1씩 증가 시켜 5행에 $i \leq 5$ 를 통해 반복회수를 정하게 된다.

7행: a를 b로 나눈 나머지를 구해서 그 값이 max와 비교해서 기존의 max가 작으면 더 큰 값으로 변경하게 된다. 5번의 데이터를 모두 입력받고 처리하면 max에는 가장 큰 나머지 값이 저장된다.

10행: max에 저장된 값을 출력하면 된다.

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1056>

⑤ 총 10개의 데이터 중 원하는 값이 들어올 경우 몇 번째 인지 출력하기

<p>상황</p>	<p>찾고자 하는 정수 값 a가 입력된다. 그리고 다음 줄부터 총 10개의 정수 값이 입력된다. 이때 a와 값이 같은 정수 값의 위치를 출력하라. 단 10개의 정수 값에는 반드시 찾고자 하는 값이 포함되어 있고 데이터의 범위는 1 ~ 100인 정수</p> <p>입력</p> <p>1 9 4 6 3</p> <p>출력</p> <p>5</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int i,value=3,temp; 4. for(i=1;i<=10;i++){ 5. scanf("%d",&temp); 6. if(temp == value) break; 7. } 8. printf("%d",i); 9. return 0; 10. }</pre>
<p>CT</p>	

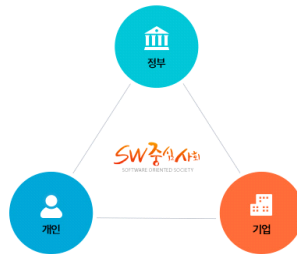
의미	<p>새로운 명령인 break가 보인다. 이와 대비되는 명령이 continue이다. break명령은 break를 둘러싼 가장 가까운 반복문을 빠져 나가게 된다. 위의 코드는 8행으로 가게 된다. continue는 실행되면 continue이하 명령문을 실행하지 않고 반복문의 증감을 입력하는 부분으로 가게 된다. 위의 코드에서는 4행에 i++로 가게 된다.</p> <p>6행: temp와 value의 값이 같은 경우 break를 하게 된다. 반복문의 중첩되어 있을 경우 break를 둘러싼 가장 가까운 반복문을 빠져 나가게 된다. 지금은 반복문이 한 개라 바로 8행으로 가게 된다. temp에 3이 입력되는 순간 반복문이 종료된다.</p> <p>8행: 종료된 순간의 i값을 출력하게 되면 찾는 위치가 된다.</p>
----	--

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1060>

SW중심대학

“ 우리 사회는 이제 SW가 개인·기업·정부의 혁신을 견인하는 SW중심사회로 진입

2015년 10대 기술트렌드 중 하나로 “이제 모든 기업은 소프트웨어 기업이다. (All companies are now software companies)” (*15.1월 뉴스위크) ”



대학교육을 SW를 중심으로 혁신함으로써, 학생·기업·사회의 SW경쟁력을 강화하고, 진정한 SW가치의 확산을 실현하는 대학

2015년 8개교, 2016년 16개

교, 2019년 20개교로 확대운영

성균관대 기존 SW학과와 컴퓨터공학과를 통합하여 SW대학을 신설(150명), '18학년도부터 총 100명 규모의 SW특기자 전형 신설·운영 (SW학과 60명, 공학계열 20명, 전자전기공학 10명 등), 우수 SW특기자에 대해서는 4년간 전액 장학금 지급

고려대 기존 유사학과(컴퓨터통신학부·컴퓨터교육과·정보보호대학원)를 통합, 국내 최대 규모의 '정보대학' 신설, '14년 학부 145명, 대학원 200명,

18학년도 컴퓨터학과 신입생 선발 시, 융합인재전형·기회균등특별전형(특성화고 졸업자), 특별전형(과학인재) 등을 통해 SW영재 12명 선발

자. 대량의 동일 데이터를 저장하는 배열 (p-197)

1) 문제 상황

총 10개의 양의 정수가 입력된다. 입력된 양의 정수를 저장한 후 다시 출력하라.

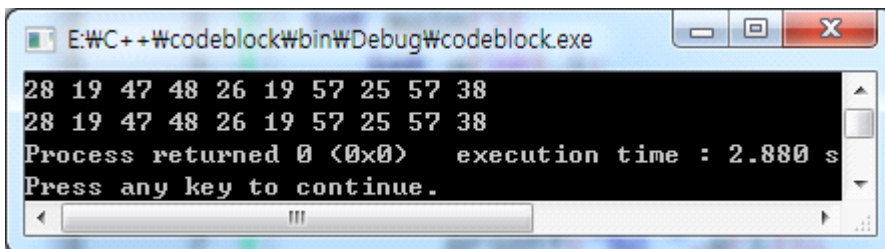
입력

28 19 47 48 26 19 57 25 57 38

출력

28 19 47 48 26 19 57 25 57 38

2) 화면



3) 소스코드

```

1. #include<stdio.h>
2. int main(){
3.     int a[10],i;
4.     for(i=0;i<10;i++)
5.         scanf("%d",&a[i]);
6.     for(i=0;i<10;i++)
7.         printf("%d ",a[i]);
8.     return 0;
9. }
```

4) 설명

많은 양의 데이터를 저장할 경우 변수를 여러 개를 사용해야 한다. 변수를 여러 개를 선언하면 되겠지만 이는 상당히 번거로운 일이다. 예를 들어 100개의 데이터를 저장하고자 한다면 a, b, c, 이러한 변수 명으로 선언해야 할 것이다. 상당히

번거로운 일이 아닐 수 없다.

따라서 이러한 문제점을 해결하기 위해 여러 개의 데이터를 동일한 이름으로 처리 할 수 있도록 배열(array)을 많은 프로그래밍 언어에서 지원한다. C언어에서는 3행과 같이 배열이름[첨자] 형태로 사용하게 된다. 이때 [] 에 입력되는 값이 처음 선언할 때 만들고자 하는 배열의 크기가 되고, 실제 사용될 때 첨자를 이용해 해당 배열의 하나의 공간에 접근하여 사용하게 된다. 그리고 항상 첫 번째 첨자는 0부터 시작하게 된다. 3행에서는 결국 a[0]부터 a[9]까지 형태의 변수 이름으로 10개가 가능하고 이러한 배열에서 첨자는 반복문을 활용하여 쉽게 접근이 가능하다.

따라서 배열 가는 곳에 반복문은 항상 존재한다고 보면 된다.

현재 위 코드는 1차원 배열이지만 다차원 배열도 만들 수 있다. 배열이름[첨자][첨자]... 식으로 만들면 된다. 무한정 배열을 만들 수 있는 것은 아니다. 이는 사용하는 컴퓨터의 메모리에 따라 다르게 된다.

따라서 위 코드는 10개의 공간을 가진 배열을 선언하고 0번부터 차례대로 데이터를 입력받아 그것을 다시 화면상에 출력하게 된다.

5) 실습예제

① 10진수를 이진수로 변환하여 출력하기

상황	<p>한 정수 a를 입력받아 2진수로 변환하여 출력하라. 단 $0 \leq a \leq 1,000$인 정수</p> <p>입력</p> <p>13</p> <p>출력</p> <p>1101</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a[100],i=0,temp; 4. scanf("%d",&temp); 5. do{ 6. a[i]=temp%2; 7. i=i+1; 8. temp=temp/2; </pre>

	<pre> 9. }while(temp>0); 10. do{ 11. i--; 12. printf("%d",a[i]); 13. }while(i); 14. return 0; 15. }</pre>
<p>CT</p>	
<p>의미</p>	<p>3행: 배열의 크기를 100으로 여유 있게 선언 6행: 2로 나눈 나머지를 배열의 0번부터 차례대로 저장 8행: 입력된 값을 2로 나눈 몫을 저장하여 반복한다. do{} while를 사용한 이유는 입력된 값이 0일 경우 0을 저장하기 위해 먼저 처리한 것이다. 11행: 배열의 첨자로 사용하는 i를 1 감소한 이유는 7행에서 나머지를 할당하고 i가 증가되어 있기 때문에 출력할 때는 1을 뺀 다음부터 감소하면서 출력해야 한다. 위의 예에서 a[0] = 13%2, a[1] = 6%2, a[2] = 3%2, a[3] = 1%2, 이때 i가 4가 되어 있기 때문에 3부터 0까지 나머지 값을 출력해야 한다.</p>

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1049>

② ‘ㄱ’모양의 배열 채우기

<p>상황</p>	<p>한 정수 n을 입력받아 1부터 n*n까지 숫자를 ‘ㄱ’모양으로 출력하라. 단 $1 \leq n \leq 10$인 정수</p> <p>입력</p> <p>3</p> <p>출력</p> <p>1 2 3 6 5 4 7 8 9</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. int a[100][100],r,n,m=1,c; 4. scanf("%d",&n); 5. for(r=0;r<n;r++){ 6. if(r%2==0){ 7. for(c=0;c<n;c++) 8. a[r][c]=m++; 9. } 10. else{ 11. for(c=n-1;c>=0;c--) 12. a[r][c]=m++; 13. } 14. } 15. for(r=0;r<n;r++){ 16. for(c=0;c<n;c++) 17. printf("%d ",a[r][c]); 18. printf("\n"); 19. } 20. }</pre>
<p>CT</p>	

의미	<p>예는 2차원 배열을 이용해서 처리한 것이다. 이차원 배열은 [][] 두 개를 사용해서 사용하게 된다. 3차원, 4차원... 등등 여러 차원은 사용자가 필요에 따라 선언하여 사용하면 된다.</p> <p>‘ㄱ’ 모양을 채울 때 유념해야 할 것은 다음 줄로 넘어 갈 경우 행의 번호가 짝수이면 열의 0번부터 채우고 행의 번호가 홀수이면 열의 n-1번부터 감소하면서 채우게 된다.</p> <p>6행: 행(r)이 짝수일 때 열(c) 0번부터 값을 채운다. 10행: 행(r)이 홀수일 때 열(c) n-1번부터 값을 채운다. 15행: 배열의 채운 내용을 차례대로 출력한다.</p>
----	--

실전문제 : <http://www.judgeon.net/JudgeOnline/problem.php?id=1066>

차. 글자는 숫자와 다르다 (p-183)

1) 문제 상황

100자 이하의 문자열이 입력된다. 다음 줄에 출력하고자 하는 글자의 수를 나타내는 정수 n 이 입력된다. 입력된 문자열에서 공백을 포함한 n 만큼의 글자를 출력하라. 단, $1 \leq n \leq 100$ 인 정수

입력

```
it is a good day to study
15
```

출력

```
it is a good da
```

2) 출력화면

```
E:\C++\codeblock\bin\Debug\codeblock.exe
it is a good day to study
15
it is a good da
Process returned 0 (0x0) execution time : 8.572 s
Press any key to continue.
```

3) 소스코드

```
1. #include<stdio.h>
2. int main(){
3.     char str[100],n;
4.     gets(str); //scanf("%s",str);
5.     scanf("%d", &n);
6.     for(int i=0;i<n;i++)
7.         printf("%c",str[i]);
8.     return 0;
9. }
```

4) 설명

먼저 4행에 ‘//’의 의미는 주석(comment)이다. 이는 프로그램의 실제 실행되는 부분은 아니고 사용자가 간단히 설명이나 표시를 위해 사용된다. 이때 //는 한 줄만 주석 처리하는 것이고 여러 줄을 주석 처리할 때에는 /* 내용 */ 형태로 사용하게 된다.

소스코드의 내용에서 먼저 문자열의 특징에 대해 이해를 해야 한다. 문자열은 문자가 하나가 아니기 때문에 저장을 할 때 char 타입을 여러 개를 이용하여 저장한다. 그리고 문자열이 입력되면 문자배열에 입력받게 된다. 예를 들어 welcome이라는 문자열을 입력받아 char str[100] 배열에 저장한다고 하면 실제 저장되는 모습이다.

0	1	2	3	4	5	6	7	8	9	.	.	.	99
w	e	l	c	o	m	e	\0						

위와 같이 char 한 칸에 하나씩의 문자가 들어간다. 그런데 문자열은 문자가 하나가 아니기 때문에 문자열의 끝을 나타내는 무언가가 있어야 한다. 이를 위해 \0 또는 W0 인 null문자를 사용하게 된다. 그래서 문자열을 만들 때에는 실제 마지막에 null문자를 위한 공간을 생각해야 한다. 그 외 8번부터는 알 수 없는 값들이 들었다. 이는 컴퓨터가 사용하는 메모리의 공간을 여러 가지 프로그램이 사용하기 때문에 메모리에 이전에 사용한 흔적이 남아 있을 수 있다. 간단히 이야기 하면 해당 프로그램이 실행되기 전에 다른 프로그램이 메모리에 값을 채우고 지우지 않기 때문이다. 보다 자세한 내용을 알고 싶은 분은 운영체제와 관련된 내용을 참고하기 바란다.

4행에서 scanf("%s",str);와 같은 형태를 사용하지 않은 이유는 scanf의 특징 때문이다. scanf는 띄어쓰기나 엔터를 하나의 입력단위로 인식하기 때문에 it is 가 입력되면 it만 받고 is는 다음 입력으로 들어오기 때문에 5행에 n에 엉뚱한 값이 입력된다.

이를 위해 gets(문자배열이름)를 이용해서 한 줄을 입력 받는 것이다. gets()(Get string from stdin)는 문자열을 입력받는 입력함수이다. 이와 반대인 문자열 출력함수인 puts(문자배열이름)도 있다. printf("%s"..)와 다른 점은 '\n'가 포함되어 있다. 출력하고 자동으로 enter를 하게 된다.

이 두 함수 역시 사용자가 직접 만들지 않았지만 사용할 수 있는 이유는 stdio.h에 포함되어 있기 때문이다. 7행에서 %c를 한 것은 문자배열에서 값을 문자로 출력

하기 위해서는 이때 %s를 하게 되면 특이한 모양으로 출력된다. 한번 변경해서 해 보길 바란다. 이는 %s는 null문자가 나올 때까지 모두 출력하기 때문이다.

유심히 보신 분들은 주석 처리된 scanf("%s",str) 이 부분에 '&'가 없다는 생각을 했을 것이다. 이는 배열의 이름은 곧 배열의 주소이기 때문이다. scanf()는 반드시 변수를 사용할 경우 주소가 와야 함으로 '&'를 붙여야 한다. 그러나 배열은 배열이름이 주소이기 때문에 붙이지 않아도 된다. 문자를 변수에 저장할 때에는 두 가지를 기억해야 한다. 연속된 공간에 한 번에 값을 입력받을 때에는 배열의 이름만 사용해야 하고 하나의 공간에 값을 입력받을 때에는 변수의 주소(&)가 와야 한다.

4) 실습예제

① 문자 반복 출력하기

<p>상황</p>	<p>100자 이하의 문자열이 입력되고 반복해야 할 수인 n이 입력된다. 이때 입력된 문자열을 n번 출력하라. 단, $1 \leq n \leq 100$인 정수</p> <p>입력</p> <p>ko 4</p> <p>출력</p> <p>ko ko ko ko</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. int main(){ 3. char str[100],n; 4. gets(str); 5. scanf("%d",&n); 6. for(int i=1;i<=n;i++) 7. printf("%s ",str); 8. return 0; 9. }</pre>
<p>CT</p>	

의미	4행: 띄어쓰기가 포함될 수 있기 때문에 gets를 사용 6행: 1부터 n번까지 7행을 반복실행 7행: str배열에 저장된 값을 %s로 문자열로 출력한다.
----	--

실전문제: <http://www.judgeon.net/JudgeOnline/problem.php?id=1310>

② 문자열의 길이를 출력하기

상황	100자 이하의 문자열이 입력될 경우 해당 문자열의 길이를 출력하라. 단, 공백도 길이로 포함하여야 한다. 입력 it is a good 출력 12
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. char str[100]; 4. gets(str); 5. for(int i=0;str[i]!='\0';i++); 6. printf("%d",i); 7. return 0; 8. }</pre>
CT	

의미	<p>5행: 반복문의 조건을 비교하는 <code>str[i]!='\0'</code>가 보일 것이다. 이는 앞에서 이야기 한 것처럼 문자열의 마지막은 null문자가 있다고 했다. 따라서 입력받은 문자열의 길이는 0부터 시작하여 null문자가 아닐 때까지 <code>i</code>를 증가시키면서 반복을 하게 되면 된다.</p> <p>이때 <code>for</code>문에 반복되어야 할 내용은 <code>i</code>를 증가 시키는 부분만 있으면 되기 때문에 다른 문장 없이 5행 마지막에 <code>;</code>(세미콜론)을 입력했다. 6행이 반복된다고 생각하면 안 된다. 그리고 <code>;</code>(세미콜론)이 없어도 될 것 같지만 반드시 반복되어야 문장이 와야 함으로 빈 공백 문장을 포함해야 한다.</p> <p>6행: 반복문의 종료되면 결국 <code>i</code>가 0부터 시작하여 null문자까지 <code>i</code>가 증가하므로 <code>i</code>를 출력해야 한다. null문자는 문자열의 길이에 포함되면 안 되기 때문이다.</p>
----	---

실전문제:<http://www.judgeon.net/JudgeOnline/problem.php?id=1312>

카. 사용자가 직접 만드는 함수 (p-283)

1) 문제 상황

양의 정수 n 을 입력받아 1부터 n 까지 합을 구하려고 한다. 추가적으로 함수를 만들어 사용해야 한다. 단, 함수의 원형은 `void add(int n);` 이고 $1 \leq n < 1,000$ 인 정수

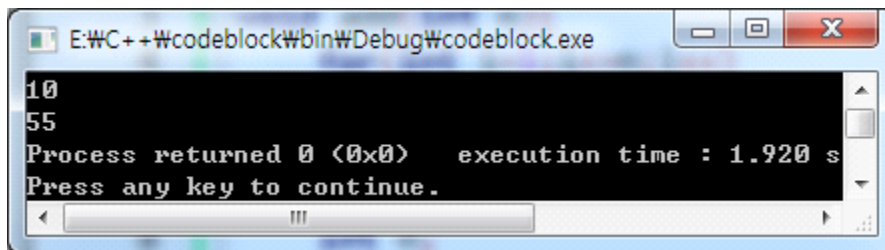
입력

10

출력

55

2) 출력화면



3) 소스코드

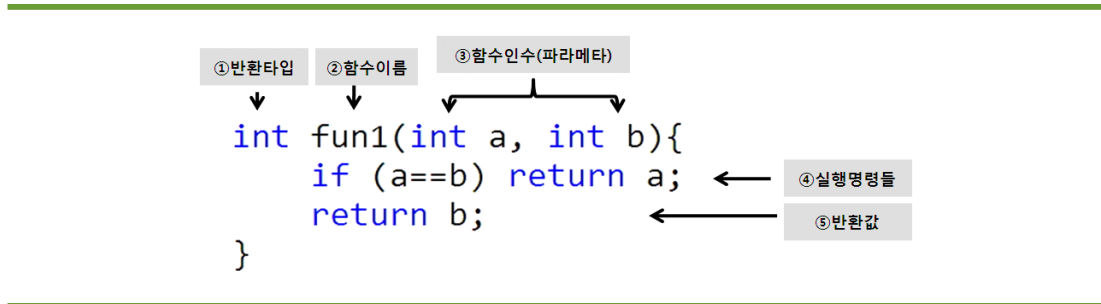
```

1. #include<stdio.h>
2. int sum;
3. void add(int n){
4.     for(int i=1;i<=n;i++)
5.         sum+=i;
6. }
7. int main(){
8.     int n;
9.     scanf("%d",&n);
10.    add(n);
11.    printf("%d",sum);
12.    return 0;
13. }

```

4) 설명

먼저 사용자 정의 함수의 형태에 대해 설명하겠다. 사용자 정의 함수는 말 그대로 사용자가 직접 함수를 정의하는 것이다. 이때 지켜야 문법이 있게 된다. 함수의 정의할 때 형태를 살펴보면,



① 함수가 끝나면서 돌려주기를 원하는 값의 타입(반환타입)이다. 일반적으로 함수의 최종 결과나 함수가 잘 실행되었는지를 알려주는 역할을 한다. 경우에 따라서 반환 값이 없는 것도 있다. 반환 값이 없다는 것을 구체적으로 표시하기 위해 void로 선언하여 사용한다. ①의 반환타입은 ⑤의 return에서 돌려주는 값의 타입이 같아야 한다.

② 함수의 이름으로 변수명과 동일한 형태로 만들면 된다.

③ 인수, 매개변수라고 부르고 실제 의미는 약간의 차이가 있다. 그러나 여기서는 같은 의미로 사용한다. 매개변수는 함수가 실행하게 될 때 필요한 내용을 호출하면서 넘겨주게 되는 값들이 된다. 매개변수도 변수이기 때문에 타입이 정의되고 정의된 타입으로 호출할 때 넘겨주어야 한다.

④ 실제 함수의 작동되는 코드가 된다.

⑤ 해당 함수가 끝나면서 돌려주는 값이 된다.

소스코드의 3행을 보면 반환(리턴)타입이 void라고 표시한 것은 함수가 끝나고 돌려주는 게 없다는 것이다. 그러나 리턴 값이 없더라도 이 함수를 실행하고 나서 값이 제대로 출력되는 것을 볼 수 있을 것이다. 리턴 값이 없더라도 변수를 이용해서 값을 계산하였기 때문에 가능하다. 이는 아래의 변수의 영역을 참고하기 바란다.

소스코드에서는 함수 이름은 add가 되고 매개변수는 한 개 이고 타입은 정수이다.

함수를 사용하기 전에 함수의 정의는 함수가 사용되기 전에 있어야 한다. 따라서 scanf(), printf(), add()함수 역시 정의는 main() 함수가 시작 전에 있게 된다. 이렇

게 작성할 경우 main()앞에 함수들이 엄청 많이 올 수도 있다. 그러면 main()함수를 사용자가 찾기도 힘들기 때문에 함수의 header만 따로 떼어 main()앞에 위치를 하고 함수의 정의 전체를 main() 밑으로 이동할 수 있다. 따라서 3행에 함수의 원형인 void add(int n); 만 적고 14행에 3행부터 6행까지를 이동해도 된다. void add(int n);을 함수의 원형이라고 한다. 이때 인수는 타입만 적고 이름은 적지 않아도 된다.

변수의 영역에 대해서 살펴보면, 변수가 참조될 수 있는 범위라고 생각하면 된다. 5행의 sum이라는 변수는 add()에는 선언되지 않았다. 따라서 add()를 둘러싸고 있는 영역에서 찾게 된다. 이러한 변수가 미치는 범위를 변수의 영역(scope)이라고 한다.

지금 예에서는 어디에도 소속되지 않는 영역 2행에서 찾게 된다. 이러한 소속되지 않는 영역을 전역(global)이라고 하고 4행에 int i=1; 8행에 int n;는 특정영역에 소속되어 있는 것을 지역(local)이라고 한다. 이에 해당하는 변수를 각각 전역변수와 지역변수라고 구분하여 부른다. 특이한 점은 전역변수는 초기화 하지 않아도 자동으로 0으로 초기화가 된다. 보다 자세한 내용은 개인적으로 찾아 보기 바란다.

※참고 : http://ko.wikipedia.org/wiki/변수_영역

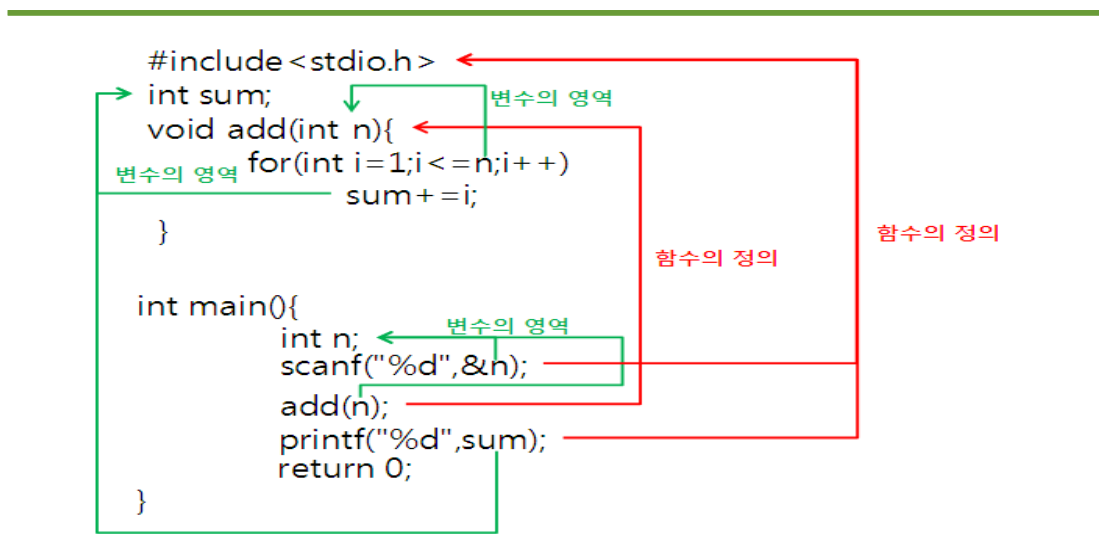
3행에 int n역시 add()속에 선언된 값이므로 지역변수가 된다. 그러면 n의 값은 어떻게 될까? 이는 함수가 호출되어지는 부분인 10행에서 값을 가져온다. 이는 함수가 호출될 때 함수의 매개변수로 n이 가진 값이 복사되어 add()에 있는 int n에게 값이 복사된다. 따라서 main()에 int n가 add()의 int n은 같은 n이라는 이름을 사용하지만 값만 복사되어 있는 전혀 다른 변수가 되는 것이다. 이는 아주 중요한 것이다. 따라서 add()에서 n의 값을 변경해도 main()에 있는 n의 값에는 영향을 미치지 않게 된다. 그러나 5행에 sum의 변화 값은 11행에서 sum을 출력할 때 변하게 된다. 이는 sum은 전역변수이기 때문에 main()과 add()에서 같이 접근하여 사용하기 때문이다. 단순히 값만 복사되어 매개변수를 통해 전달되는 것을 값에 의한 전달(call by value)이라고 한다. 이외에 참조에 의한 전달(call by reference)도 있지만 이는 나중에 포인터에 가서 자세히 다루도록 하겠다. C언어에 대해 좀 더 깊이 배우려고 하는 분들은 기타 C언어 교재에서 포인터 부분을 참고하기 바란다.

위 소스 코드는 7행인 main()부터 실행하여 9행에서 n을 입력받고 10행에서 사용자 정의 함수를 호출하여 매개변수로 n의 값만 복사되어 add()를 실행하게 된다.

add()함수는 stdio.h에 없기 때문에 함수의 정의가 있는 곳으로 이동하게 된다. 이때 3행에 int n에 main()에서 받은 n의 값이 복사되어 전달된다. 이를 바탕으로 4, 5행에서 sum에 합을 누적하게 된다. 이때 sum은 add()함수에 있는 것이 아니라 전역 변수이기 때문에 sum의 값이 변하게 된다. 6행에서 add()함수는 종료되고 다시 호출한 main()로 돌아오게 된다. 그러면 11행인 sum을 출력한다.

이때 sum은 main()에 없기 때문에 바깥 영역에 있는 2행에서 찾게 된다. 이때 영역의 구분은 { }로 보면 된다.

아래 그림이 다소 복잡하지만 이번 예제의 전부를 간단히 표현 한 것이다. 아래의 내용만 제대로 숙지한다면 함수에서 많은 부분을 이해하리라 본다.



5) 실습예제

※ 함수를 반드시 이용해서 풀어야 하는 문제는 없다. 따라서 기존에 나온 문제를 함수를 이용해 풀어보기 바란다.

- ① 변수의 영역에 따른 출력결과

상황	변수의 영역에 따른 출력결과를 이해하시오. 입력 없음 출력 아래의 화면 참고
----	--

<p>코드</p>	<pre> 1. #include<stdio.h> 2. int a=10; 3. int f(int a){ 4. a=25; 5. printf("%d\n",a); 6. return a; 7. } 8. int main(){ 9. printf("%d\n",a); 10. int a=20; 11. f(a); 12. printf("%d\n",a); 13. }</pre>
<p>CT</p>	
<p>의미</p>	<p>9행: a는 10행의 a를 참조할 수 없다. 따라서 바깥 영역인 2행에 있는 a를 출력한다.</p> <p>11행: a에 20을 값을 복사해어 3행 int a에게 넘기게 된다.</p> <p>5행: 출력되는 것은 4행에 있는 25가 출력되고 6행에서 리턴 값을 돌려준다.</p> <p>12행: 이때 a는 10행에 있는 20이 출력된다.</p>

② n개의 과목을 입력받아 평균을 출력하는 함수

<p>상황</p>	<p>과목의 수인 n이 입력된다. 다음 줄에 n개 과목의 점수가 입력된다. 입력된 과목의 평균을 함수를 통해 계산해 보자. 단, 평균은 소수 세 번째에서 반올림해서 두 번째까지 표시한다. $1 \leq n \leq 10$인 정수이다.</p> <p>입력</p> <p>4</p> <p>10 40 60 90</p> <p>출력</p> <p>36.67</p>
<p>코드</p>	<pre> 1. #include<stdio.h> 2. double aver(int n,int sum){ 3. return sum/(double)n; 4. } 5. int main(){ 6. int i, n,temp,sum=0; 7. scanf("%d",&n); 8. for(i=1;i<=n;i++){ 9. scanf("%d",&temp); 10. sum+=temp; 11. } 12. printf("%.2lf",aver(n,sum)); 13. return 0; 14. }</pre>
<p>CT</p>	

의미	<p>2행: 함수의 리턴 타입은 실수이고 정수 인수 2개를 넘겨받는다.</p> <p>3행: 인수 둘 다 정수이기 때문에 하나를 강제로 형을 변환하여 실수 연산이 되도록 한다.</p> <p>6행: 합을 저장하기 위한 sum=0으로 초기화하여 쓰레기 값이 저장되지 않도록 한다.</p> <p>8행~11행: n개의 과목의 점수를 누적해서 sum에 더한다.</p> <p>12행: aver함수를 호출하면서 과목의 개수 n 과 합계 sum의 값을 인자로 넘겨주고 함수의 리턴 값을 받아 소수 두 번째 까지 반올림하여 출력한다.</p>
----	--

③ 주어진 수의 약수의 개수를 반환하는 함수

상황	<p>양의 정수 n이 입력된다. 이때 n의 약수의 개수를 출력하라. 단, $1 \leq n \leq 1,000,000$인 정수</p> <p>입력</p> <p>6</p> <p>출력</p> <p>4</p>
코드	<pre> 1. #include<stdio.h> 2. void cnt(int n){ 3. int i=1, count=0; 4. for(i=1; i*i<=n;i++){ 5. if(n%i==0){ 6. if(i*i==n) count++; 7. else count+=2; 8. } 9. } 10. printf("%d",count); 11. } 12. int main(){ 13. int n; 14. scanf("%d",&n); 15. cnt(n); </pre>

	<pre>16. return 0; 17. }</pre>
CT	
의미	<p>2행: 함수를 호출한 곳으로 어떠한 값도 돌려줄 필요가 없기 때문에 void로 선언하여 사용하였다.</p> <p>3행: 약수의 개수를 저장하기 위한 count를 변수를 선언한다.</p> <p>4행: 1부터 시작해서 $i*i \leq n$까지 약수의 개수를 센다. 이는 실제로는 n까지 i를 증가하면서 차례대로 나눠야 이는 시간이 다소 오래 걸린다. 하지만 n이 i로 나눠지면 n/i도 약수가 된다. 따라서 $i*i$까지만 비교를 해도 약수의 개수를 구할 수 있다.</p> <p>5행~7행: 완전제곱수인 4, 9, 16, 25, ... 등은 $2*2$, $3*3$, $4*4$, $5*5$는 약수를 하나로 처리해야 때문에 이를 구분하기 위해 제어문을 이용해 구분한다.</p> <p>14행: main함수에서는 정수 n만 입력받는다.</p> <p>10행: 최종 약수의 개수를 출력한다.</p> <p>15행: 입력받은 정수 n을 인자로 해서 cnt()함수를 호출한다. 이때 이 함수를 통해 main쪽에서 하는 것은 호출만 하고 그 외에는 필요 없다.</p>

타. 여러 가지 데이터를 하나로 구조체 만들기 (p-277)

1) 문제 상황

학생에 대한 정보가 입력된다. 이름, 나이, 국어성적, 영어성적, 수학성적이 입력된다. 이를 바탕으로 학생의 이름, 나이, 성적합계, 평균을 출력하려고 한다. 단, 이름은 영문자 3글자, 나이는 정수, 성적은 정수, 평균은 실수로 셋째자리에서 반올림하여 둘째자리 까지 표시한다.

입력

```
kbs 35 90 70 85
```

출력

```
kbs 35 245 81.67
```

2) 출력화면

```
E:\C++\codeblock\bin\Debug\codeblock.exe
kbs 35 90 70 85
kbs 35 245 81.67
Process returned 0 (0x0) execution time : 7.441 s
Press any key to continue.
```

3) 소스코드

```
1. #include<stdio.h>
2. struct stuinfo{
3.     char name[10];
4.     int age, kor, eng, math;
5.     int sum;
6.     double avg;
7. };
8. double average(int sum, int cnt){
9.     double temp=(double)sum/cnt;
10.    return temp;
```

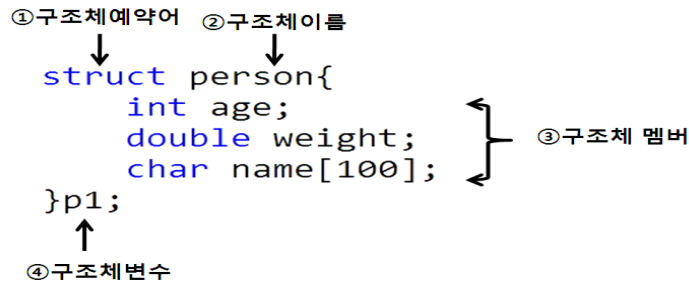
```

11. }
12. int main(){
13.     struct stuinfo s1;
14.     scanf("%s%d%d%d",s1.name, &s1.age,&s1.kor,&s1.eng,&s1.math);
15.     s1.sum = s1.kor+s1.eng+s1.math;
16.     s1.avg=average(s1.sum,3);
17.     printf("%s %d %d %.2lf",s1.name, s1.age, s1.sum, s1.avg);
18.     return 0;
19. }

```

3) 설명

문제를 풀다 보면 하나의 변수타입이 아니라 다양한 타입을 한 번에 처리해야 할 때가 있다. 이때 사용하는 것이 struct를 이용한 구조체를 만드는 것이다. struct는 이미 정의된 변수타입을 활용하여 사용자가 임의로 하나의 타입으로 묶는다고 보면 된다. 이는 배열과의 차이점이다. 물론 구조체는 같은 타입을 하나로 묶어도 된다.



구조체 변수를 사용하기 위해 사용자는 자신이 사용할 변수를 모아서 구조체를 선언해야 한다. ①과 같이 구조체 예약어인 struct로 시작하고 ②와 같이 구조체 이름을 정의를 해야 한다. 이 이름은 앞에 배운 변수의 타입과 같은 형태로 사용된다. ③은 하나로 묶을 구조체의 멤버들이 된다. ④는 선언된 구조체를 통해 p1이라는 변수를 하나 만든 것이다. 이것은 구조체 선언과 동시에 구조체 변수를 하나 만든 것이다. 반드시 필요한 것은 ;(세미콜론)은 있어야 되지만 p1는 없어도 된다. 왜냐하면 구조체 변수는 필요할 때 선언해서 사용하면 된다.

소스코드를 살펴보면 다음과 같다.

2행부터 7행까지 구조체에 대한 선언이다. 구조체 선언이 어디에도 소속되지 않는 전역영역에 한 이유는 다른 함수에도 사용하기 위함이다. 8행부터 11행까지는 평균을 반환하는 함수 average를 만든 부분이다.

13행에 보면 구조체 변수를 선언한다. 그 이름은 s1이다. 이는 일반적인 변수선언과 동일한 형태다 단지 struct stuinfo가 있는 것뿐이다. 14행보면 특이하게 구조체 변수 다음에 .(점)이 나온다. 이는 구조체의 멤버를 접근하기 위한 연산자이다. 따라서 구조체로 선언된 변수의 멤버를 접근하기 위해서는 .(점)연산자를 사용해야 접근이 가능하다. 이외에는 일반적인 변수와 동일하게 사용하면 된다.

15행은 국어, 영어, 수학의 합계를 계산한다. 16행은 합과 과목의 개수를 함수의 인수를 넘겨 평균을 돌려받고 17행에서 각 값을 출력하게 된다.

9행에 (double)은 평균을 실수 값이기 때문에 강제로 형을 변환하기 위한 것이다.

5) 실습예제

① 가장 큰 좌표 구하기

상황	<p>입력은 총 n개의 x,y로 이루어진 좌표가 입력된다. 이때 x 값이 가장 큰 것과 y 값이 가장 큰 것의 좌표를 각 줄에 출력하시오. 단 $1 \leq n \leq 100$인 정수, $-1,000 \leq x, y \leq 1,000$인 정수</p> <p>입력</p> <p>3 2 4 5 9 8 7</p> <p>출력</p> <p>8 7 5 9</p>
코드	<pre> 1. #include<stdio.h> 2. struct Point{ 3. int x, y; 4. }data[101]; 5. int main(){ 6. int n,i,xmax=0,ymax=0; 7. scanf("%d",&n); 8. data[0].x=-1000; 9. data[0].y=-1000; 10. for(i=1;i<=n;i++){ </pre>

	<pre> 11. scanf("%d%d",&data[i].x,&data[i].y); 12. if(data[0].x < data[i].x) { 13. data[0].x=data[i].x; 14. xmax=i; 15. } 16. if(data[0].y < data[i].y){ 17. data[0].y =data[i].y; 18. ymax=i; 19. } 20. } 21. printf("%d %d\n",data[xmax].x,data[xmax].y); 22. printf("%d %d\n",data[ymax].x,data[ymax].y); 23. return 0; 24. }</pre>
<p>CT</p>	
<p>의미</p>	<p>2행: 좌표를 저장하기 위한 Point구조체 선언하고 구조체배열을 101개 선언하다. 이는 0번 좌표를 x의 최대와 y의 최대를 입력하기 위해 사용한다.</p> <p>6행: xmax, ymax는 구조체 배열 몇 번 첨자에 가장 큰 값이 입력되는지 기억하기 위해 사용된다.</p> <p>8행,9행은 좌표 중 가장 작은 -1000을 입력해 둔다.</p> <p>11행: 좌표를 하나씩 입력받는다.</p> <p>12행: x좌표 중에 가장 큰 값을 저장하기 위해 입력된 값과 비교한다. 13행 큰 x좌표를 넣어두고 14행 해당 위치를 저장한다.</p> <p>16행: y좌표 중에 가장 큰 값을 저장하기 위해 입력된 값과 비교한다. 13행 큰 y좌표를 넣어두고 14행 해당 위치를 저장한다.</p> <p>21행,22행: 기억된 첨자를 이용해 x좌표의 큰 값과 y좌표의 큰 값을 각각 출력한다.</p>

실전예제 : <http://www.judgeon.net/JudgeOnline/problem.php?id=1350>

과. 거울속의 거울 (p-283)

1) 문제 상황

1부터 n까지 합을 재귀함수를 통해 계산해 보자. 단 n은 1000보다 작은 양수이다.

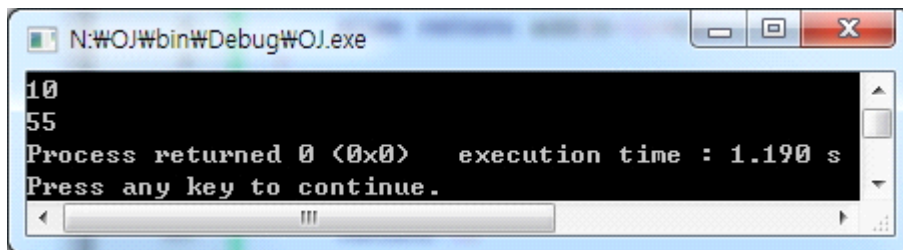
입력

10

출력

55

2) 출력화면



3) 소스코드

```

1. #include<stdio.h>
2. int add(int n){
3.     if(n==1) return 1;
4.     else return add(n-1)+n;
5. }
6. int main(){
7.     int n;
8.     scanf("%d",&n);
9.     printf("%d",add(n));
10.    return 0;
11. }

```


4) 설명

재귀(Recursion)는 원래의 자리로 되돌아가거나 되돌아옴을 뜻한다. 재귀의 특성은 여러 가지 반복이나 특별히 반복되는 패턴들을 만들어 내는 특성을 가지고 있으며, 프랙탈 구조도 이러한 특성을 가지고 있다. 이를 바탕으로 프로그래밍 언어에서는 재귀함수를 만들 수 있다.

재귀함수에서 재귀적 정의가 필요하다. 이는 자신을 재 참조하는 것을 의미한다.

예를 들어 함수 $f(n)$ 을 1~ n 까지의 합을 구하는 함수라고 정의하자. 여기서 $f(n)$ 을 재귀적으로 정의하기 위해서 $f(n-1)$ 을 이미 안다고 가정하면 함수 $f(n)$ 은 다음과 같이 정의 할 수 있다.

$$f(n) = \begin{cases} 1 & (n = 1) \\ f(n-1) + n & (n > 1) \end{cases}$$

이와 같이 재귀적 함수를 만드는 과정은 수학의 증명 방법 중 하나인 수학적 귀납법과 느낌이 비슷하다.

수학적 귀납법은 수학에서 어떤 식이 성립함을 증명하기 위해 사용되는 방법이다. 수학적 귀납법 차원에서 살펴보면 $f(1)=1$ 은 첫 번째, $f(n)=f(n-1)+n$ 은 임의의 n 에 대해서 $f(n-1)$ 을 알고 있다면 $f(n)=f(n-1)+n$ 이 성립한다. 위와 같이 재귀적 정의로 구성된 식을 수학에서는 점화식이라고 한다. 이러한 점화식을 만들 수 있다면 이를 바로 재귀함수로 작성 할 수 있다.

수학에서의 점화식은 수를 기반으로 구성된다. 하지만 정보과학에서는 수뿐만 아니라 모양, 이미지 등의 모든 객체에 대해서 점화식으로 표현할 수 있다.

5) 실습예제

- ① 1부터 n 까지 홀수의 합을 재귀함수로 계산해 보자

상황	입력은 n 이 입력되고 이때 1부터 n 까지 합을 재귀함수를 이용해 계산해 보자. 단 n 은 1000보다 작은 양의 정수이다. 입력 5 출력 9
----	--

코드	<pre> 1. #include<stdio.h> 2. int odd(int n){ 3. if(n==1) return 1; 4. else{ 5. if(n%2==0) return odd(n-1); 6. else return odd(n-1)+n; 7. } 8. } 9. int main(){ 10. int n; 11. scanf("%d",&n); 12. printf("%d",odd(n)); 13. return 0; 14. }</pre>
CT	
의미	<p>3행 : n이 1이 되면 종료되어야 하기 때문에 1을 리턴하고 끝난다.</p> <p>5행 : n이 1이 아닐 경우는 n이 짝수일 경우는 홀수가 되도록 n-1을 호출하고, 홀수일 경우 그 값을 더하고 n-1을 호출한다.</p> <p>결국 5가 입력될 경우 $f(4)+5=f(3)+5=f(2)+3+5=1+3+5$가 된다.</p> <p>따라서 결과는 $1+3+5$가 되어 9를 출력하게 된다.</p>

실전예제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=1284>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1281>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1282>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1041>

하. 스택(stack)이 뭐죠? (p-202)

1) 문제 상황

수식이 주어질 때 매치되는 괄호를 찾는 문제이다. 예를 들어 $(3*(4+4)+1)$ 일 경우 시작위치가 0부터 시작한다고 하면 3과 7, 0과 10이 한쌍의 괄호이다.

입력

최대 100자인 수식이 공백없이 입력된다.

입력예시 : $(3*(4+4)+1)$

출력

앞에서부터 짝이 맞는 순으로 출력한다. 단, 괄호 짝이 맞지 않을 때에는 'not match'를 출력한다.

출력예시 :

3 7

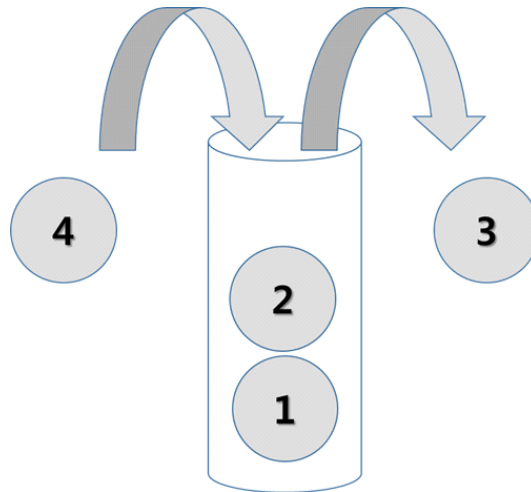
0 10

CT

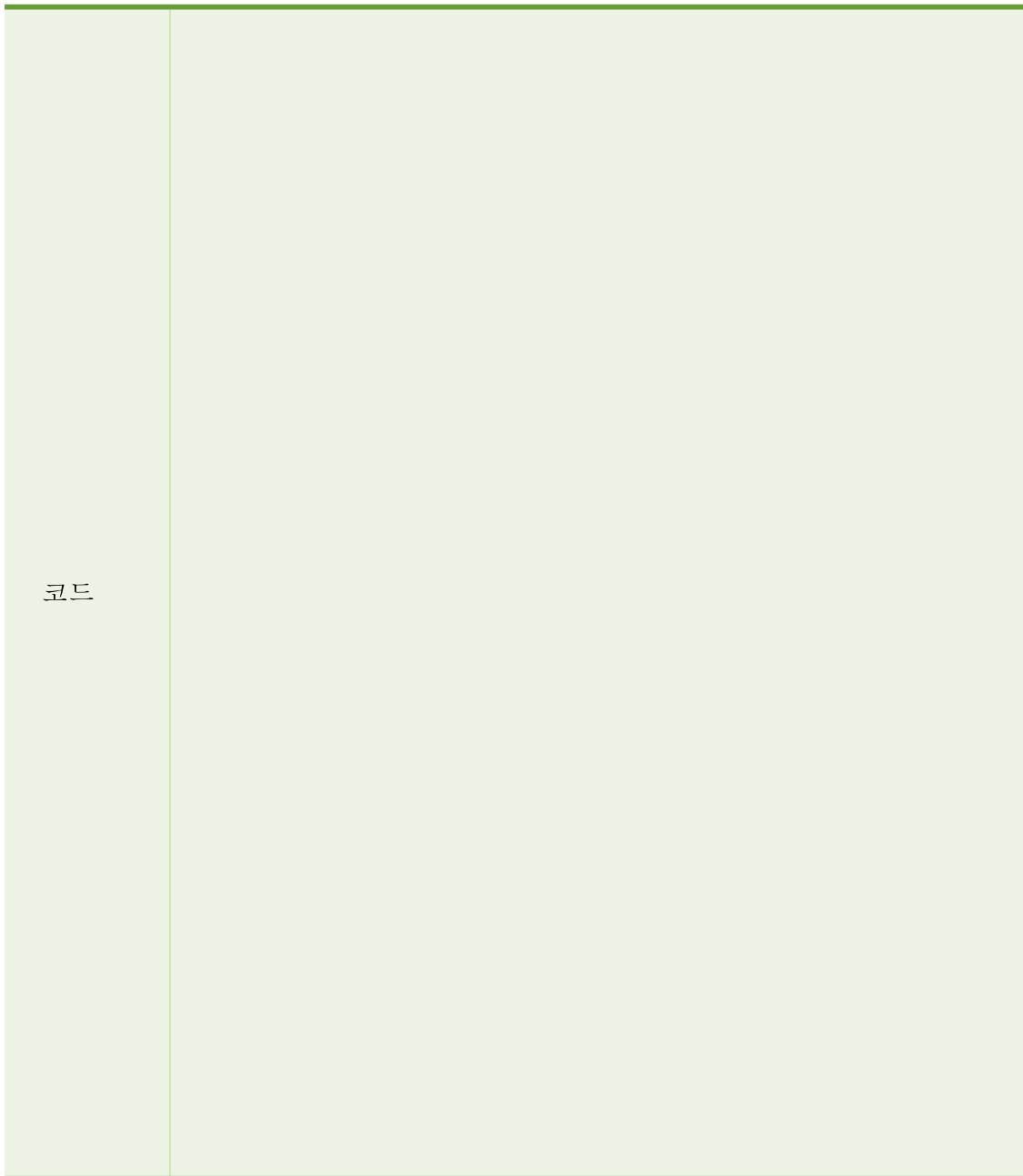
2) 설명

컴퓨터에서 자료를 저장할 때 자료를 효율적으로 사용하기 위해서는 자료의 특성에 따라서 분류하여 저장하고 관리해야 가능하다. 이러한 자료를 관리하기 위한 구조와 연산을 묶어서 자료구조(data structure)라고 한다.

자료구조는 아주 다양하게 많이 있다. 그 중에서 스택(Stack)에 대해서 알아보자. 스택은 자료를 한 쪽이 막힌 통에 쌓아(stack) 올린 형태의 구조이다. 따라서 볼 수 있는 자료는 항상 마지막에 있는 자료이고 다음 자료를 보기 위해서는 가장 위에 있는 것을 꺼내야 가능하다. 스택이 가능한 연산은 크게 삽입(push), 삭제(pop)가 있다. 이와 더불어 현재 스택이 비워있는지(isEmpty), 스택이 가득차 있는지(isFull), 현재 스택에 가장 위에 있는 값이 무엇인지의 연산이 필요하게 된다.



3) 스택의 구현



실전예제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=2010>

<http://www.judgeon.net/JudgeOnline/problem.php?id=2011>

<http://www.judgeon.net/JudgeOnline/problem.php?id=2014>

<http://www.judgeon.net/JudgeOnline/problem.php?id=5031>

<http://www.judgeon.net/JudgeOnline/problem.php?id=8006>

<http://www.judgeon.net/JudgeOnline/problem.php?id=1321>

가. 큐(Queue)는 뭐죠? (p-202)

1) 문제 상황

숫자와 -(빼기기호)로 구성된 문자열이 입력된다. 이때 - 기호가 나오면 먼저 입력된 숫자 하나를 삭제한다. 마지막에 남아 있는 값들을 출력하시오.

입력

데이터의 개수 $N(1 \leq N \leq 100)$ 이 입력되고 입력되는 숫자는 0~9이다.

입력예시 : 9 5 9 - 3 2 8 - - 1

출력

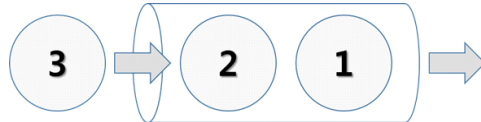
남아 있는 값 출력하라. 단 - 기호가 입력될 때 지울 수 있는 숫자가 없을 경우 error를 출력하라.

출력예시 : 2 8 1

CT

2) 설명

큐(Queue)는 양쪽이 뚫린 원통을 생각하면 된다. 한쪽에서는 입력되고 다른 한쪽은 출력된다고 보면 된다. 따라서 항상 먼저 입력된 자료가 먼저 나오게 된다. 가능한 연산은 크게 삽입(push), 삭제(pop)가 있다. 이와 더불어 현재 큐가 비워있는지(isEmpty), 큐가 가득차 있는지(isFull), 큐의 마지막의 값의 위치와 가장 앞에 있는 값의 위치가 필요하다.



3) 큐의 구현

CT	
코드	

실전예제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=3021>

나. 문제해결과정 1 (p-239)

1) 문제 상황

가장 적은 수의 동전으로 원하는 금액을 거스려 주려고 한다. 가지고 있는 동전은 500원, 100원, 50원, 10원, 1원 5가지이다. 530원을 돌려주려면 최소 동전의 개수는?

입력

거스름돈의 액수(최대 10000원)

입력예시 : 530

출력

최소 동전의 수

출력예시 : 4

CT

2) 실습예제

① 거스름돈 II : <http://www.judgeon.net/JudgeOnline/problem.php?id=2704>

CT

② 거스름돈 III: <http://www.judgeon.net/JudgeOnline/problem.php?id=2705>

CT

다. 문제해결과정 2 (p-243)

1) 문제 상황

7에서 시작해 가장 아래까지 숫자를 하나씩 선택해 내려갈 때, 가장 큰 숫자 합을 만들 있는 방법과 그 합은 무엇일까? 단, 하나의 숫자를 선택하면 아래 숫자는 바로 아래 좌우 2개의 숫자에서만 선택할 수 있다. 예를 들어 7-3-1-5-2-5의 순서대로 선택해 내려가면 합이 23이 된다.

			7						
			3		8				
		8		1		4			
	2		6		5		4		
	3		6		2		7		1
7		7		6		5		5	
									7

입력

높이 N이 입력된다.

N줄에 걸쳐 불링핀의 점수가 차례대로 주어진다.

입력예시:

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

출력

최대점수를 출력한다.

2) 문제풀기

CT	
----	--

3) 실습예제

- ① 광석수집 : <http://www.judgeon.net/JudgeOnline/problem.php?id=2803>

CT	
----	--

- ② 파스칼의 삼각형: <http://www.judgeon.net/JudgeOnline/problem.php?id=1210>

CT	
----	--

라. 문제해결과정 3 (p-247)

1) 문제 상황

1*2 크기의 타일을 이용하여 2*n크기의 타일을 채우는 방법의 경우의 수를 구하라

입력

타일의 크기 n이 입력된다.($1 \leq n \leq 100,000$)

입력예시 : 1

출력

경우의 수를 출력한다.

출력예시 : 1

2) 문제풀기

CT	
----	--

3) 실습예제

① 타일 채우기 II : <http://www.judgeon.net/JudgeOnline/problem.php?id=3202>

CT	
----	--

하. 오일러 프로젝트를 시작해 보자

사이트 주소

- 영문 : <http://projecteuler.net/> 한글 : <http://euler.synap.co.kr/>

1) 오일러 프로젝트란?

오일러 프로젝트는 수학적/컴퓨터프로그래밍 문제들을 풀어내기 위한 시리즈 문제들로, 문제를 풀어내기 위한 수학적 통찰 이상을 필요로 합니다. 비록 수학적 기술들은 문제를 풀어내기 위한 근사하고 효율적인 방법들을 만들어내는데 도움을 줄 수 있겠지만, 컴퓨터와 프로그래밍 언어 사용능력은 대부분의 문제를 풀어내는데 꼭 필요할 것입니다.

2) 어떤 사람들을 대상으로 하고 있는가?

일반적인 교육과정에서 배움의 욕구를 채우지 못한 학생들을 포함한, 여러 사람들, 원래 수학적 배경지식이 없지만 수학적 사고에 관심을 가지고 있는 성인들, 그리고 최고의 경지에서 그들의 문제 풀이와 수학적 능력을 계속적으로 유지하고 싶은 전문가들.

3) 누구나 문제를 풀 수 있나?

문제들은 난이도에 따라서, 그리고 계속적으로 반응을 유도하는 연속적인 많은 학습경험들로 배치되어 있습니다. 따라서, 한 문제를 풀어내면 새로운 것을 알게 되고 그럼으로써 이전에는 가까이 하기에 매우 어려운 문제들에 도전해 볼 수 있도록 한다. 따라서 열심히 참여하는 사람들은 매 문제를 풀 때 마다 느리지만 확실히 향상될 것이다.

4) 며칠씩 걸려서 답을 얻을 수 있어도 될까요?

모든 문제들은 1분 규칙이 적용되어있기 때문에, 좀 더 어려운 문제를 확실히 풀어내기 위해 여러 가지 알고리즘을 고민해야 한다.

6) 실습예제

① 배수의 합 계산

상황	<p>10보다 작은 자연수 중에서 3 또는 5의 배수는 3, 5, 6, 9이고, 이것을 모두 더하면 23이다. 그러면 1000보다 작은 3 또는 5의 배수의 합은 얼마인가?</p> <p>입력</p> <p>1000</p> <p>출력</p> <p>233168</p>
코드	<pre> 1. #include<stdio.h> 2. int main(){ 3. int n,i,sum=0; 4. scanf("%d",&n); 5. for(i=1;i<n;i++) 6. if(i%3==0 i%5==0) sum+=i; 7. printf("%d",sum); 8. return 0; 9. }</pre>
CT	
의미	<p>5행: n은 포함이 되지 않기 때문에 i<n으로 작성</p> <p>6행 : 3의 배수 또는 5의 배수 이기 때문에 (or)연산을 사용하고 둘 중에 하나에 해당될 경우 합을 계산한다.</p>


실전예제 :

<http://www.judgeon.net/JudgeOnline/problem.php?id=2600>

<http://www.judgeon.net/JudgeOnline/problem.php?id=2601>

<http://www.judgeon.net/JudgeOnline/problem.php?id=2602>

<http://www.judgeon.net/JudgeOnline/problem.php?id=2604>

 이상으로 C언어 아주 기초적인 부분을 다루었다. 이 정도문법으로도 처음 프로그래밍을 공부하는 분들은 충분할 것이다. 조금 더 깊이 공부를 하고 싶은 분들은 관련 서적을 찾아 참고하면 되겠다.

3. Online judge site

가. Online Judge site란?

온라인 저지 사이트(Online Judge Site:일명 OJ)는 온라인에서 사용자가 코드를 제출하면 해당 서버에서 사용자 코드를 실행하여 서버에 있는 데이터 값과 비교해서 사용자에게 제출한 프로그램의 통과여부를 알려주는 사이트를 말한다. 이때 사용자에게 프로그램의 크기, 실행속도, 메모리 사용량, 프로그램 통과여부, 시간초과, 컴파일에러, 불법적 접근등의 메시지를 알려주게 된다.

나. 국내외 Online Judge site

1) 국외 Online Judge site

① 미국 : USACO - <http://www.usaco.org/>

USA Computing Olympiad

OVERVIEW TRAINING CONTESTS HISTORY STAFF RESOURCES

JANUARY CONTEST
The USACO 2015 January contest has recently ended. Results are [here](#).

OUR MISSION
The USACO supports computing education in the USA and worldwide by identifying, motivating, and training high-school computing students at all levels. We provide:

- Hundreds of hours of free on-line [training resources](#) that students can use to improve their programming and computational problem-solving skills.
- On-line [programming contests](#) (roughly six per year) for students at all levels.
- An intensive summer [training camp](#), to which the top students in the USA are invited to further improve their skills and learn advanced material.
- The opportunity for the top four students in the USA to represent their

YOUR ACCOUNT
Not currently logged in.
Username:
Password: [Forgot password?](#)

2014-2015 SCHEDULE
Dec 12-15: First Contest
Jan 16-19: Second Contest
Feb 20-23: Third Contest
April 3-6: US Open
May 21-30: Training Camp
July 26-August 2: IOI'15 in Almaty, Kazakhstan

연습할 수 있는 사이트 training도 제공하고 실제 대회도 진행하고 있다. 현재 2014~2015까지 대회 일정이 대충 나와 있다. 본 사이트에서는 Bronze, Sliver, Gold로 등급이 나뉘져 있고 처음에는 Bronze에서 출발해서 Gold로 실력을 쌓아 올라가게 된다.

② 러시아 : Codeforce - <http://codeforces.com/>

The screenshot shows the Codeforces website interface. At the top, there is a navigation bar with links for HOME, CONTESTS, GYM, PROBLEMSET, RATING, HELP, and several tournament icons. The main content area features a post titled "2013-2014 CT S01E11: Codeforces Trainings Season 1 Episode 11" by MikeMirzayanov. A code snippet is displayed in a dark box:

```
int getRandomNumber()
{
    return 4; // chosen by fair dice roll.
             // guaranteed to be random.
}
```

Below the code, there is a caption: "그림. codeforces 웹사이트 첫 화면". To the right of the main content, there are side panels including a "Pay attention" section for an upcoming contest and a "Top rated" table with columns for rank, user, and rating.

- 특이한 점은 대회가 시작되면 사용자가 제출한 코드를 다른 사람이 분석해서 점수를 획득할 수 도 있는 특이한 시스템으로 구성된 온라인 저지 사이트이다. 물론 종료된 대회의 문제는 누구나 도전해서 풀 수 있다. 엄청 자주 대회를 실시하는 사이트로 해외 각국의 우수한 정보인재들과 겨뤄볼 수 있는 아주 좋은 사이트 중 하나이다.

③ 폴란드: <http://www.spoj.com>

The screenshot shows the SPOJ website. At the top, there is a blue header with the SPOJ logo and the text "Sphere online judge". Below the header, there is a login form with fields for "User:" and "Password:", a checkbox for "for today", and a "Log In" button. To the right of the login form, there is a large promotional banner with the text "Geeks rule the WORLD - if you agree please VOTE and win trip to Cebit 2014!". Below the banner, there is a table showing statistics for SPOJ in other languages:

SPOJ in other languages:	Armenian	Polish	Portuguese (BR)	Vietnamese		
10514079 total submissions	225641 registered users	4513 public problems	17191 all problems	3416 affiliated institutions	3071 organized contests	48 programming languages

At the bottom of the screenshot, there is a timestamp and a user comment: "2013-09-02 23:00:44 Do you teach programming? by Łukasz Kuszner".

그림. spoj 웹사이트 첫 화면

④ 중국 : www.poj.org



그림. poj 웹사이트 첫 화면

⑤ 일본 : <http://judge.u-aizu.ac.jp/onlinejudge/>

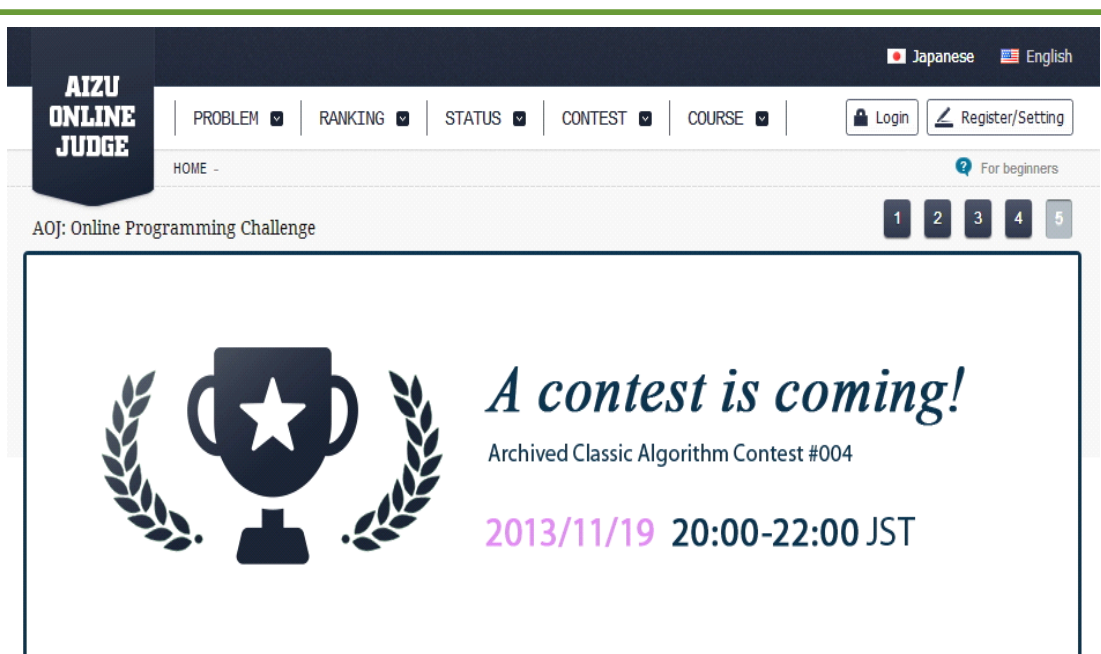


그림. aizu online judge 웹사이트 첫 화면

2) 국내 Online judge Site

① <http://www.koistudy.net>



- 경기과학고 정종광 선생님이 운영, 모의고사를 수시로 실시하고 주로 경기과학고 학생들이 활동함, 회원승인 후 사용가능

② <http://codeup.kr>



- 경남정보고 배준호 선생님이 운영, 주로 전국단위 처음 배우는 학습자에게 적합, 자유롭게 가입 후 사용가능

③ <http://www.judgeon.net>



- 경남과학고 김봉석 선생님이 운영, 주로 경남과학고 학생들이 사용, 자유롭게 가입 후 사용가능

④ <http://koi.codingfun.net/usage>

이름	버전	날짜
KOIDataPull	0.84	2012.04.29
KOIValidator	0.85	2012.04.29
SimpleKOIValidator	0.98.1.7	2013.07.13

- 충북과학고 및 충북지역 학생들 이용, 자유롭게 가입 후 사용가능

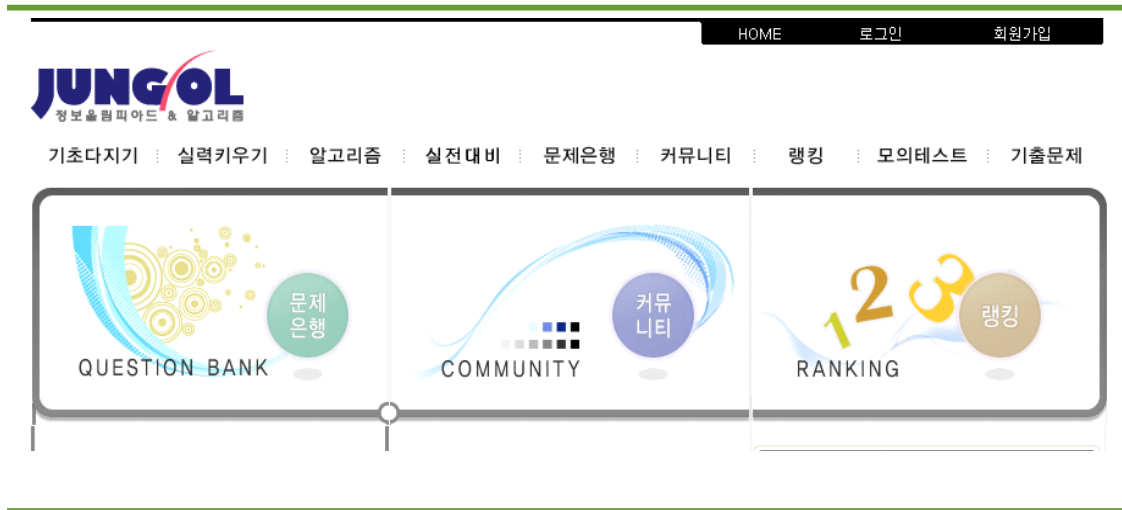
⑤ <http://www.acmicpc.net/>

- 대학생 프로그래밍 경진대회를 위한 사이트, 일반인들에게 공개 됨, 다양한 프로그래밍 언어 지원(C, C++, Ruby, pascal, python, php...)

⑥ <http://algotspot.com/>

- 주로 대학생 및 프로그래밍에 관심이 많은 일반인들이 활동, 상당히 난이도 있고 문제들이 우수함. 초보자에게는 어려울 수 있음.

⑦ <http://www.jungol.co.kr/site/>



- 사단법인에서 운영, 무료로 사용 가능, 모의고사를 자주 개최

⑧ <http://www.dovelet.com>

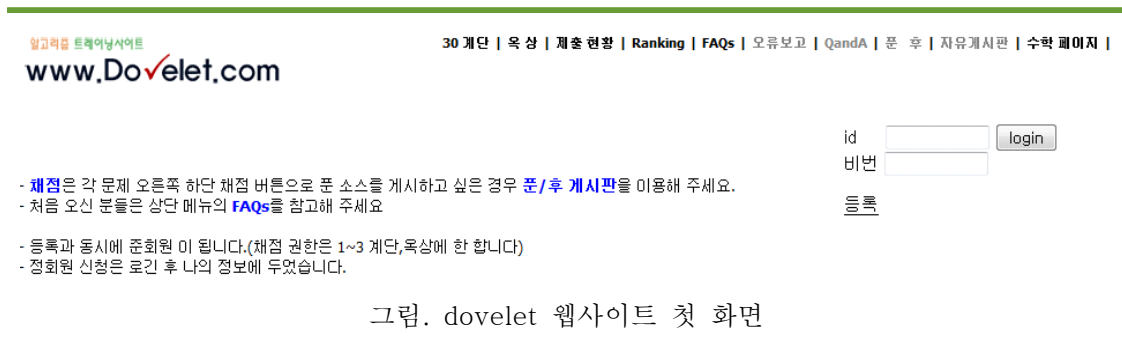


그림. dovelet 웹사이트 첫 화면

- 개인이 운영하는 사이트, 모든 문제를 풀기 위해서는 랭킹 안에 들거나 일부 비용을 지불해야 함.

! 소개된 사이트 외에도 다양한 온라인 시스템을 제공하고 있다. 관련 기업들이 실시하는 대회 등도 많이 있다. 이러한 사이트가 중요한 것은 일회성이 아니라 자신의 실력을 수시로 체크하고 점검하여 실력을 꾸준히 쌓을 수 있다는 것이다. 프로그래밍을 통한 실력은 단기간에 되지 않는다. 여러분들도 꾸준히 소개한 사이트에서 활동 한다면 자신의 실력이 향상되고 있음을 느낄 것이다.