



ROBIT
ROBOT SPORT GAME TEAM

C Language 기초 – 1일차

1. 기초 이론
2. 변수, 연산자
3. 조건문, 반복문



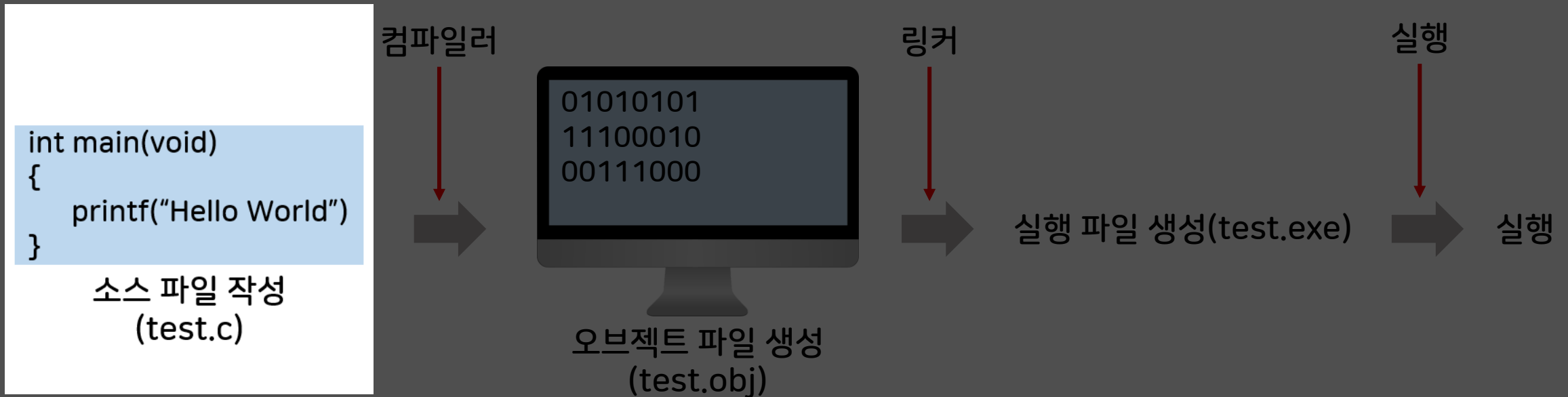
ROBIT
ROBOT SPORT GAME TEAM

기초 이론

프로그래밍 실행 절차



ROBIT
ROBOT SPORT GAME TEAM



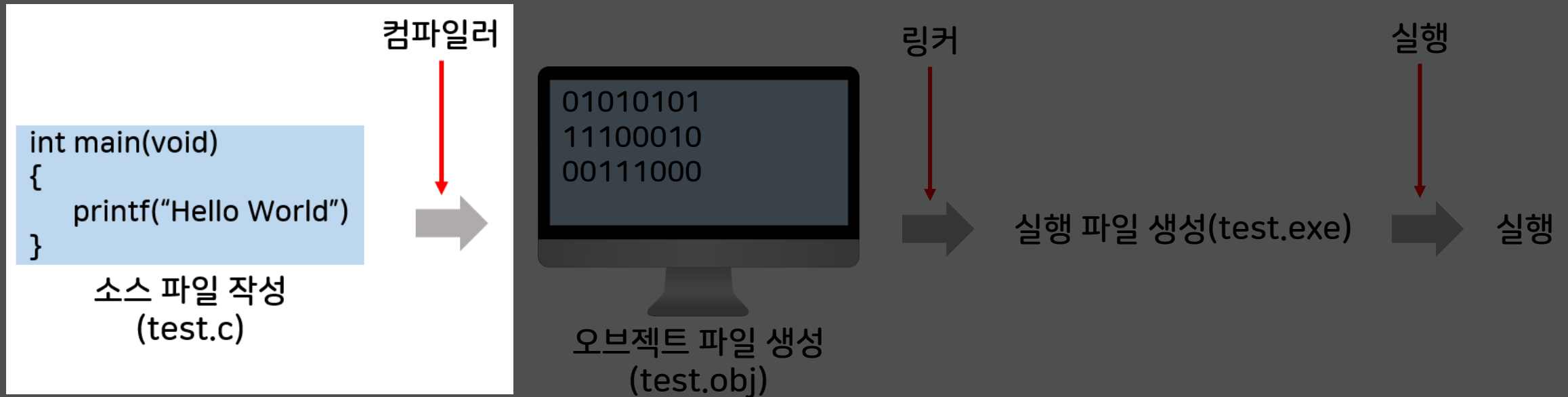
소스파일 (source file) 이란?

프로그래밍 언어를 이용하여 원하는 작업의 내용을 기술한 코드가 들어있는 파일
C언어의 소스 파일의 확장자는 항상 ".c" 여야 한다.

프로그래밍 실행 절차



ROBIT
ROBOT SPORT GAME TEAM



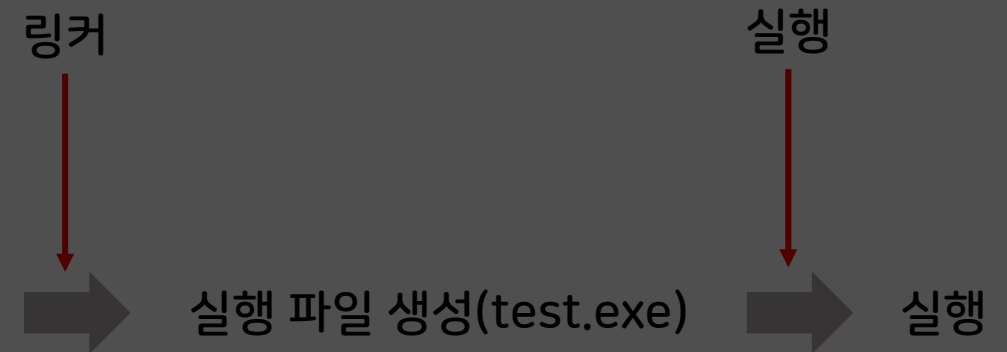
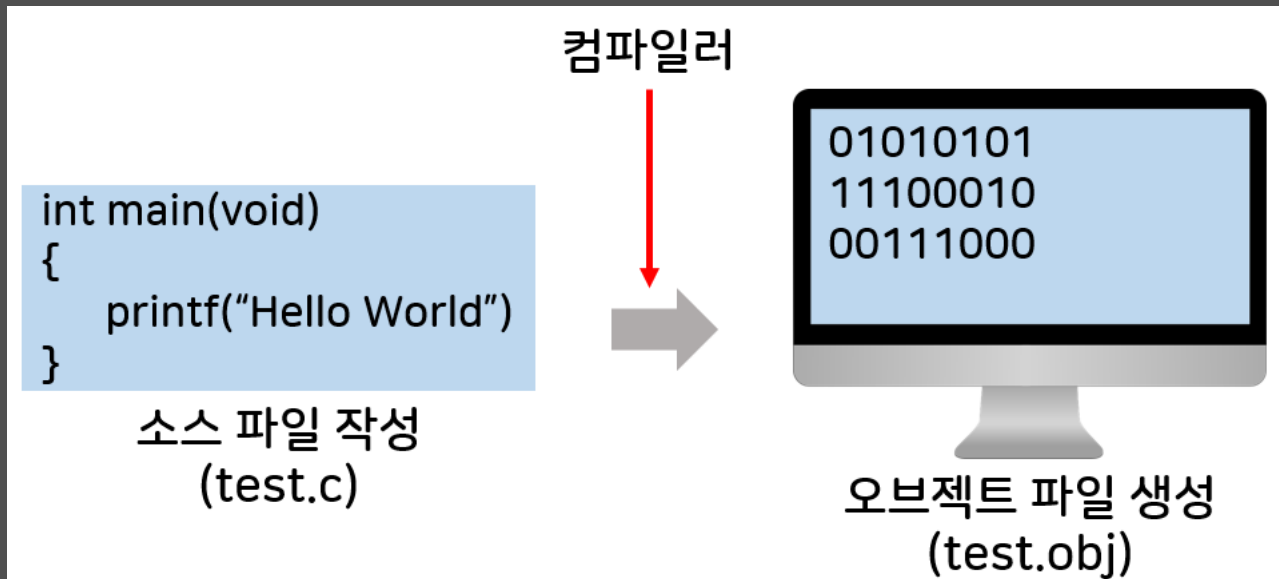
컴파일러 (compiler) 란?

고급언어문장(C,C++ 등)을 컴퓨터가 이해할 수 있는 기계어 코드로 바꾸는 작업을 하는 프로그램

프로그래밍 실행 절차



ROBIT
ROBOT SPORT GAME TEAM



오브젝트 파일 (object file) 이란?

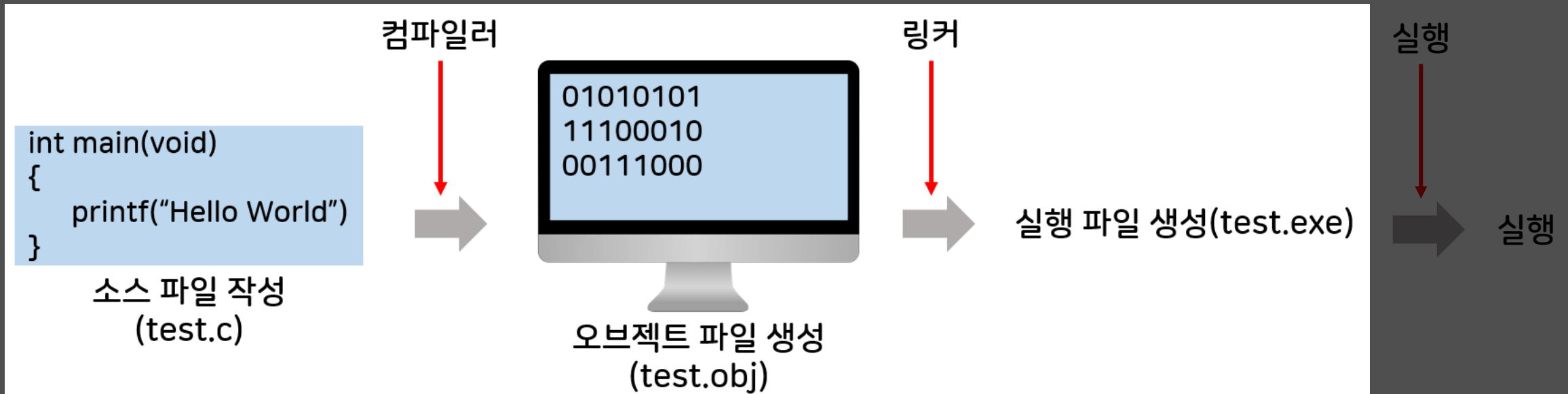
컴파일러를 통해 소스코드가 기계어로 변환되고, 변환된 기계어로 구성되어 있는 파일을 오브젝트 파일이라고 한다.
windows에서는 확장자가 ".obj" 이다.

(test.c 를 컴파일하면 test.obj 파일이 생성된다.)

프로그래밍 실행 절차



ROBIT
ROBOT SPORT GAME TEAM



링커 (linker) 이란?

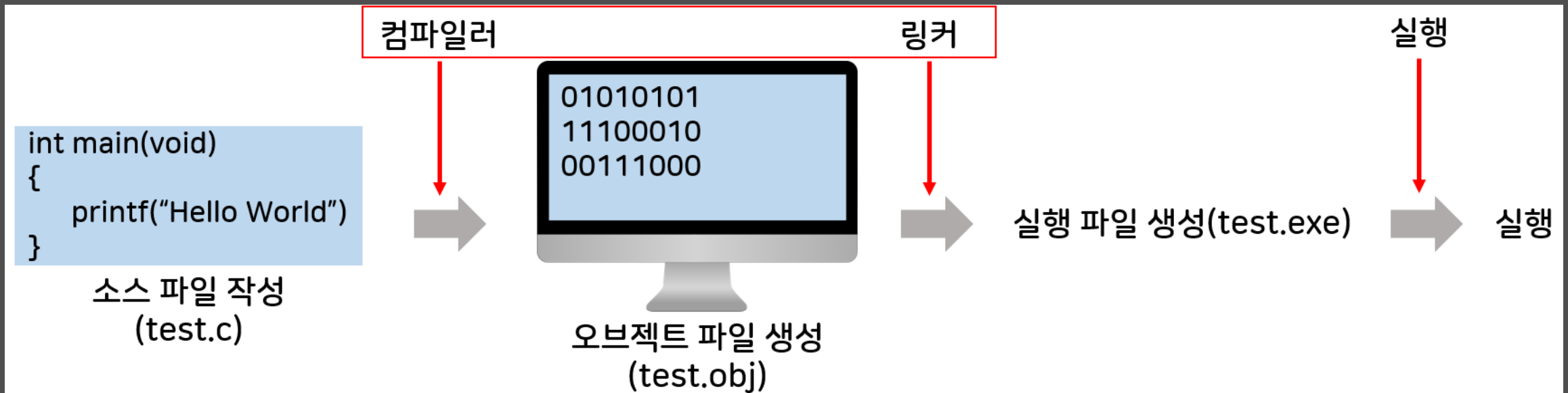
오브젝트 파일들을 라이브러리와 연결하는 링크과정을 수행하고 실행 파일을 생성하는 동작을 한다.
확장자는 ".exe" 이다.

프로그래밍 실행 절차



ROBIT
ROBOT SPORT GAME TEAM

빌드(build)



이와 같은 프로그래밍 실행 절차를 통해 원하는 동작을 컴퓨터로 실행할 수 있게 된다.
Visual Studio에서는 컴파일러와 링크과정을 묶어 빌드(build)라고 한다.

비트와 바이트



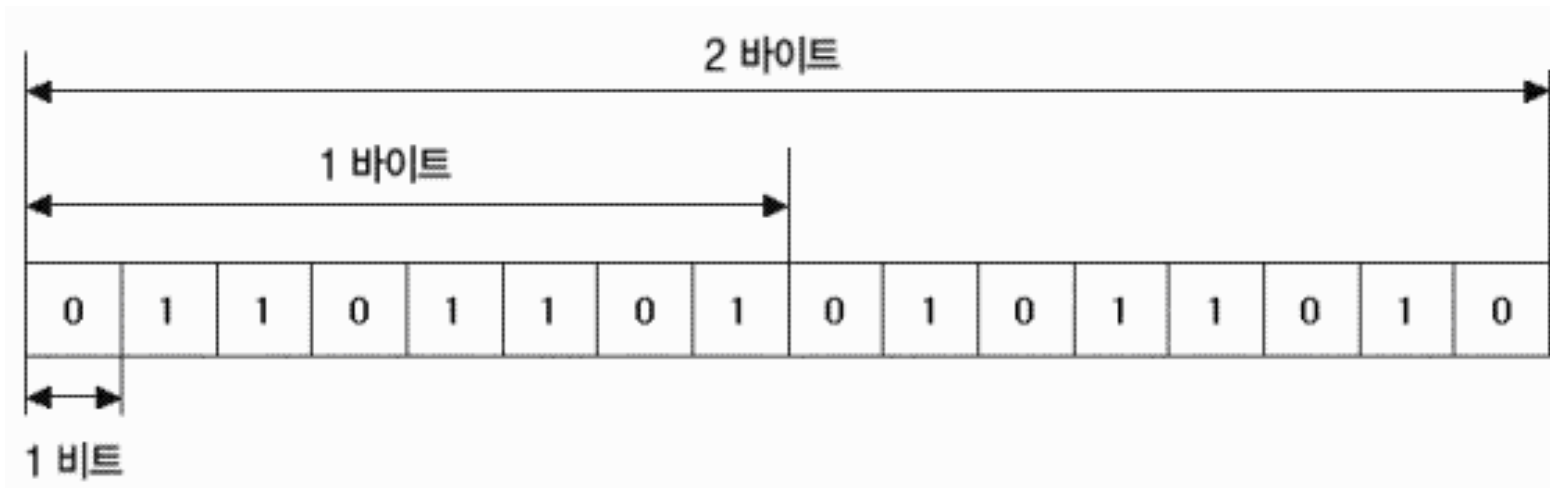
ROBIT
ROBOT SPORT GAME TEAM

1. 비트(bit)란 무엇인가?

컴퓨터는 0과 1만을 사용하는 이진수로 표현한다.
이러한 이진수 하나의 자릿수를 비트라 한다.

2. 바이트(byte)란 무엇인가?

8개의 비트를 모은 것을 바이트라고 한다.





ROBIT
ROBOT SPORT GAME TEAM

변수(Variable) 와 자료형(data type)



변수(Variable)

- 변수란 무엇인가?

프로그램에서 데이터를 저장하는 일시적인 메모리 공간

- 변수는 왜 필요한가?

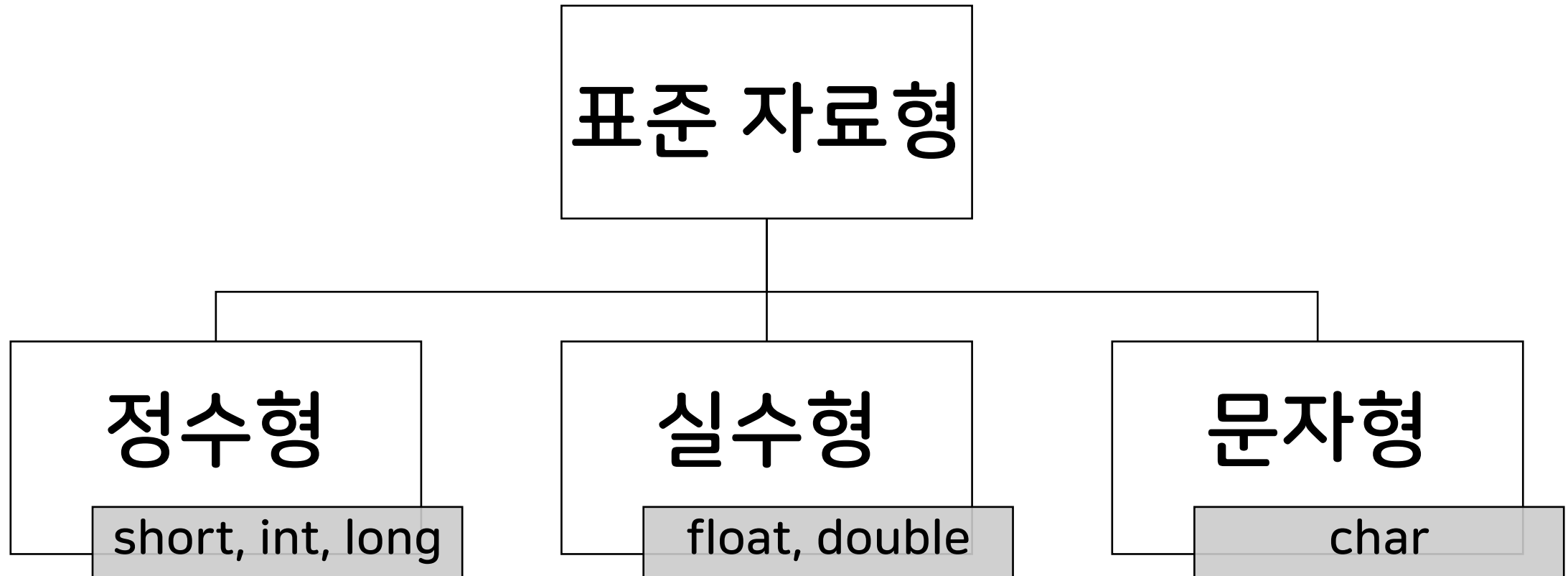
데이터가 입력되면 어딘가에 저장해야만 다음에 사용할 수 있다.

- 데이터를 메모리 공간에 저장할 때, 유형(type)을 구별하여 저장한다.
- 변수를 구별하는 유형을 '**자료형**'(data type)이라 한다.

자료형(data type)



ROBIT
ROBOT SPORT GAME TEAM



자료형의 종류



ROBIT
ROBOT SPORT GAME TEAM

자료형			설명	바이트 수	범위
정수형	부호 있음	short	short형 정수	2	$-2^{15} \sim (2^{15})-1$
		int	정수	4	$-2^{31} \sim (2^{31})-1$
		long	long형 정수	4	$-2^{31} \sim (2^{31})-1$
	부호 없음	unsigned short	부호 없는 short형 정수	2	$0 \sim (2^{16})-1$
		unsigned int	부호 없는 정수	4	$0 \sim (2^{32})-1$
		unsigned long	부호 없는 long형 정수	4	$0 \sim (2^{32})-1$
문자형	부호 있음	char	문자 및 정수	1	$-2^7 \sim (2^7)-1$
	부호 없음	unsigned char	문자 및 부호 없는 정수	1	$0 \sim (2^8)-1$
부동 소수점형		float	단일 정밀도 부동 소수점	4	$1.2\text{E}-38 \sim 3.4\text{E}38$
		double	두배 정밀도 부동 소수점	8	$2.2\text{E}-308 \sim 1.8\text{E}308$

자료형의 크기

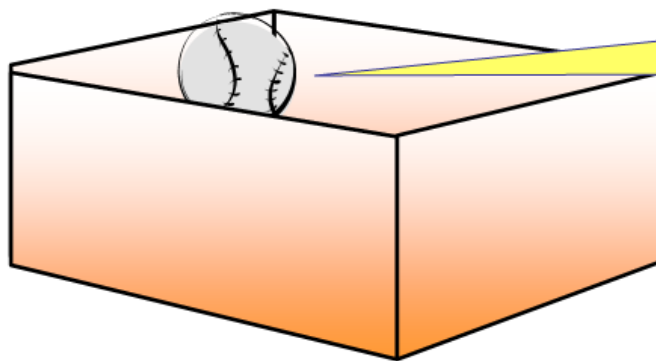


자료형의 필요성

- 메인 메모리의 효율성



물건이 상자보다 크면 들어가지 않을 것이다.

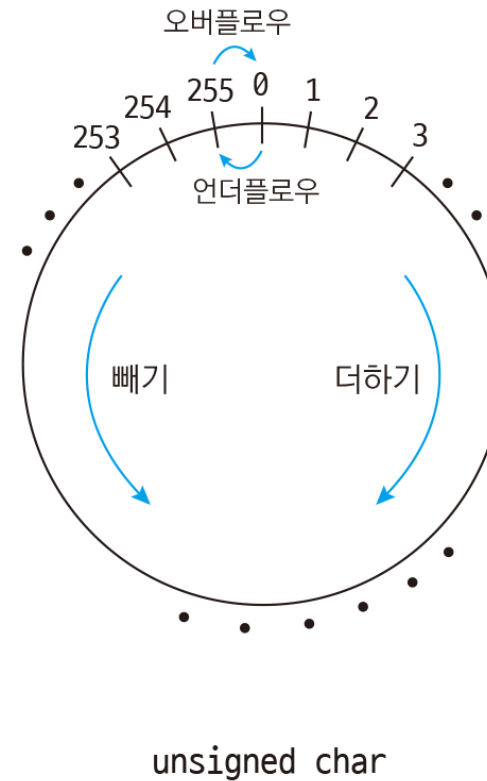
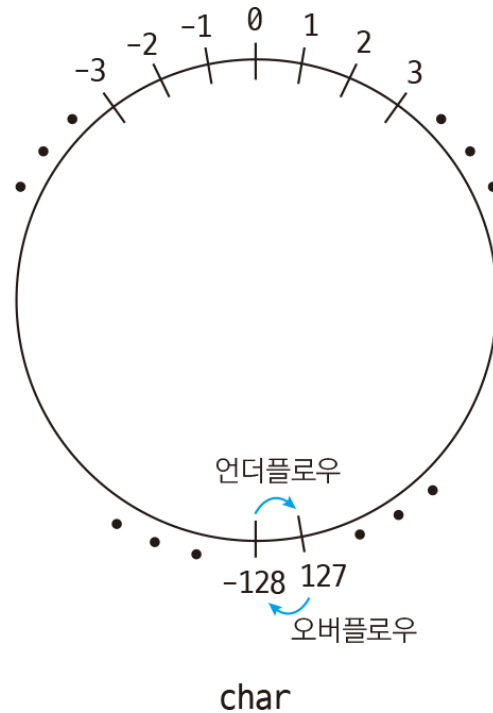


물건이 상자보다 너무 작으면 공간이 낭비될 것이다.

오버플로우, 언더플로우



ROBIT
ROBOT SPORT GAME TEAM



허용 범위를 넘는 값을 변수에 저장할 경우
사용자가 의도한 값이 아닌 전혀 다른 값이 저장되는 현상
컴파일러는 오류로 받아들이지 않음

변수 선언



ROBIT
ROBOT SPORT GAME TEAM

- 변수 선언 : 컴파일러에게 어떤 변수를 사용하겠다고 미리 알리는 것

자료형 변수이름 = 초기 값;

- 변수 선언의 예

char c = 'A';

int i = 0;

double ratio = 0.0;

변수(Variable)



ROBIT
ROBOT SPORT GAME TEAM

변수의 타입	변수의 이름	주소 값	데이터 값
char	A	4FFCF3 번지	'A'
int	i	4FFCE4 번지	0
float	ratio	4FFCD8 번지	0.0

변수는 메인 메모리에 만들어져서
각 변수들은 주소 값을 가진다

변수의 값을 저장하는 방법



ROBIT
ROBOT SPORT GAME TEAM

```
char c = 'A';
```

```
int i = 0;
```

```
double ratio = 0.0;
```

```
// 문자형 변수 c 선언과 초기화
```

```
// 정수형 변수 i 선언과 초기화
```

```
// 실수형 변수 ratio 선언과 초기화
```

```
c = 'b';
```

```
i = 200;
```

```
ratio = 2.78;
```

```
// 변수 c에 문자 'b'를 대입
```

```
// 변수 i에 200을 대입
```

```
// 변수 ratio에 2.78을 대입
```



변수의 이름 짓기

- 식별자(identifier) :

변수 이름, 함수 이름 등에 사용되어서 다른 이름들과 식별할 수 있게 해주는 것

- 식별자를 만드는 규칙

1. 알파벳 문자와 숫자, 밑줄 문자(_)로 구성
2. 첫 번째 문자는 반드시 알파벳 또는 밑줄 문자(_)
3. 대문자와 소문자는 구별된다.
4. C 언어 키워드와 똑같은 이름은 허용하지 않는다.

변수의 이름 짓기



ROBIT
ROBOT SPORT GAME TEAM

Ex)

sum	0	
_count	0	
adcValue	0	
2nd_try	X	// 숫자로 시작
\$dollar	X	// \$ 기호로 시작
double	X	// C 언어 키워드

변수의 이름 짓기 Tip



ROBIT
ROBOT SPORT GAME TEAM

변수의 역할을 가장 잘 설명하는 이름을 지어야 한다.

ex)

bankAccount
number
sum



ROBIT
ROBOT SPORT GAME TEAM

printf()

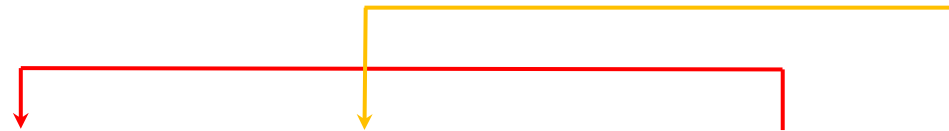
printf()



- printf()
 - 모니터에 출력을 하기 위한 표준 출력 라이브러리 함수
 - " " 안의 내용을 그대로 모니터에 출력 (문자열 출력)
 - 변환기호에 맞춘 형태로 모니터에 출력 (변수 출력)

%d	정수 형태로 출력	100
%f	실수 형태로 출력	3.141592
%c	문자 형태로 출력	A
%s	문자열 형태로 출력	Hello

- printf("학번 %d의 성적은 %f 입니다", number, grade);



printf()



ROBIT
ROBOT SPORT GAME TEAM

표 3-3 변환기호의 종류

변환기호	설명
%d	인자의 내용을 부호 있는 10진수로 변환하여 출력한다.
%o	인자의 내용을 부호 없는 8진수로 변환하여 출력한다.
%x	인자의 내용을 부호 없는 16진수로 변환하여 출력한다.
%u	인자의 내용을 부호 없는 10진수로 변환하여 출력한다.
%e	인자의 내용을 지수 형태로 출력한다(부동소수점 형태).
%f	인자의 내용을 실수 형태로 출력한다(부동소수점 형태).
%g	인자의 내용을 %e 또는 %f의 변환 중 문자 수가 적은 것으로 출력하며 유효하지 않은 0은 출력되지 않는다.
%c	인자의 내용을 문자 1개로 인식하여 출력한다.
%s	인자의 내용을 문자열로 인식하여 출력한다.
%n	%n이 나타날 때까지 출력한 문자 수를 해당 인자에 저장한다.
%ld	인자를 long형 10진수로 출력한다.
%lu	인자를 long형 부호 없는 10진수로 변환하여 출력한다.

변환기호(명세)

제어 문자열에 변환기호를 추가하면 대응하는 값을 해당 형식에 맞게 출력

printf 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int Number = 65;
06     printf("65를 8진수로 변환 : %o\\n", Number);
07     printf("65를 10진수로 변환 : %d\\n", Number);
08     printf("65를 16진수로 변환 : %x\\n", Number);
09     printf("65를 문자형으로 변환 : %c\\n", Number);
10
11     return 0;
12 }
```

```
C:\\Program Files\\Microsoft Visual Studio\\MyProjects\\ss\\Debug\\...
65를 8진수로 변환 : 101
65를 10진수로 변환 : 65
65를 16진수로 변환 : 41
65를 문자형으로 변환 : A
```


예제 1



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int num1 = 10, num2 = 5;
06
07     printf("num1의 초깃값은 %d이다.\n", num1);
08     printf("num2의 초깃값은 %d이다.\n", num2);
09
10     // num2의 값에 새로운 정숫값을 저장한다.
11     num2 = 20;
12     // num1의 값에 새로운 정숫값을 저장한다.
13     num1 = 30;
14
15     printf("num1의 새로운 값은 %d이다.\n", num1);
16     printf("num2의 새로운 값은 %d이다.\n", num2);
17     return 0;
18 }
```

```
C:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
num1의 초깃값은 10이다.
num2의 초깃값은 5이다.
num1의 새로운 값은 30이다.
num2의 새로운 값은 20이다.
Press any key to continue
```

예제2



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     float a = 12.34;
06     double b = 56.78;
07
08     printf("float형 a의 값은 %f이다.\n", a);
09     printf("double형 b의 값은 %lf이다.\n", b);
10
11     // 소수점 자릿수를 조정하면
12     printf("소수점 자릿수를 조정한 후의 값\n");
13     printf("float형 a의 값은 %.2f이다.\n", a);
14     printf("double형 b의 값은 %.3lf이다.\n", b);
15
16     return 0;
17 }
```

```
c:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
float형 a의 값은 12.340000이다.
double형 b의 값은 56.780000이다.
소수점 자릿수를 조정한 후의 값
float형 a의 값은 12.34이다.
double형 b의 값은 56.780이다.
Press any key to continue
```



ROBIT
ROBOT SPORT GAME TEAM

scanf()



scanf()

- 표준 입력장치인 키보드로부터 입력된 데이터를 지정된 형식으로 변환하여 변수에 저장하는 라이브러리 함수이다.

- scanf() 함수의 기본 형식

```
scanf("변환기호(Format-String).....", &변수1, &변수2, .....);
```

값을 저장할 변수의 주소

- 변환기호와 변수의 관계

```
scanf("%d", &age);
```

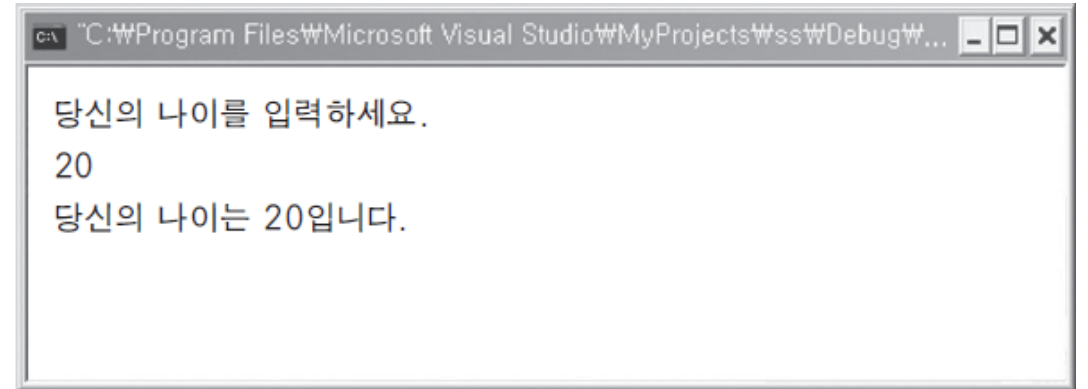
- 변수 앞에 변수의 주소를 연산하는 기호인 &를 반드시 넣어야 함.

scanf 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int age = 0;
05
06     printf("당신의 나이를 입력하세요.\n");
07     scanf_s("%d", &age);
08     printf("당신의 나이는 %d입니다.\n", age);
09
10     return 0;
11 }
```





ROBIT
ROBOT SPORT GAME TEAM

상수(Constant)



상수(Constant)

- 변수(Variable) :
한번 값이 저장되었어도 언제든지 다시 다른 값으로 변경이 가능하다.
- 상수(Constant) :
값이 한번 정해지면 변경할 필요가 없는 데이터들을 상수로 사용한다.
상수는 프로그램이 실행되는 동안 값이 변경되지 않는다.



상수(Constant)

- 기호 상수(symbolic constant) : 상수에 이름을 부여한 것

ex) π = 3.14159.....

- 기호 상수 사용법

1. # define 문장 사용
2. const 키워드 사용

ex) # define PI 3.14159

ex) const int PI = 3.14159

기호상수 값

↑ ↑

- 기호 상수의 필요성

1. 프로그램의 가독성을 높여줌
2. 유지보수가 쉽다

상수 예제



ROBIT
ROBOT SPORT GAME TEAM

```
#include <stdio.h>
#define PI 3.141592    // 기호 상수 정의

int main(void)
{
    float radius, area, circumference;    // 변수 선언

    radius = 4.0;
    area = PI * radius * radius;          // 면적 계산
    circumference = 2.0 * PI * radius;    // 둘레 계산

    printf("반지름은 %f입니다.\n", radius); // 반지름 출력
    printf("원의 면적은 %f이고 둘레는 %f입니다.\n", area, circumference);

    return 0;
}
```



ROBIT
ROBOT SPORT GAME TEAM

연산자



기능에 따른 연산자의 분류

연산자의 분류	연산자	의미
대입	=	오른쪽을 왼쪽에 대입
산술	+ - * / %	사칙연산과 나머지 연산
부호	+ -	부호를 나타낸다
증감	++ --	증가, 감소 연산
관계	> < == != >= <=	오른쪽과 왼쪽을 비교
논리	&& !	논리적인 AND, OR
조건	?	조건에 따라 선택
coma	,	피연산자들을 순차적으로 실행
비트 단위 연산자	& ^ ~ >> <<	비트별 AND, OR, XOR, 이동, 반전
sizeof 연산자	sizeof	자료형이나 변수의 크기를 바이트 단위로 반환
형변환	(data type)	변수나 상수의 자료형을 변환
포인터 연산자	* & []	주소 계산, 포인터가 가리키는 곳의 내용 추출
구조체 연산자	. ->	구조체의 멤버 참조

연산자 우선순위



ROBIT
ROBOT SPORT GAME TEAM

표 4-12 연산자 우선순위

우선순위	연산자의 종류		연산자	결합성
고 ↑ ↓ 저	식, 구조체, 공용체		(), [], .., →	좌 → 우
	단항 연산자		!, ~, -, +, ++, --, &, *	좌 → 우
	이항 연산자	승제	*, /, %	좌 → 우
		가감	+, -	
		이동	<<, >>	
		비교	<, <=, >, >=	
		등가	==, !=	
		비트 AND	&	
		비트 XOR	^	
		비트 OR		
		논리 AND	&&	
		논리 OR		
	조건 연산자		?:	좌 ← 우
	대입 연산자		=, +=, -=, *=, /=, <<=, >>=, &=	좌 ← 우
	콤마 연산자		,	좌 → 우

산술 연산자



ROBIT
ROBOT SPORT GAME TEAM

- 나머지 연산자 (%)

10 % 2는 0 이다.

5 % 7는 5 이다.

30 % 9는 3 이다.

- 증감 연산자

++x : x값을 먼저 증가한 후에 다른 연산에 사용한다.

x++ : x값을 먼저 사용한 후에 증가한다.

--x : x값을 먼저 감소한 후에 다른 연산에 사용한다.

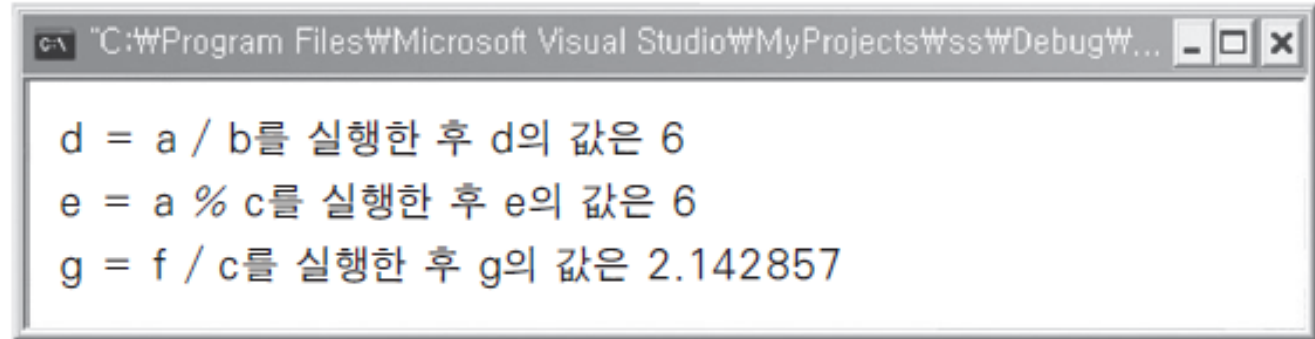
x-- : x값을 먼저 사용한 후에 감소한다.

산술 연산자 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a = 20, b = 3, c = 7, d, e;
06     float f = 15.0, g;
07
08     d = a / b;
09     printf("d = a / b를 실행한 후 d의 값은 %d\n", d);
10
11     e = a % c;
12     printf("e = a %% c를 실행한 후 e의 값은 %d\n", e);
13
14     g = f / c;
15     printf("g = f / c를 실행한 후 g의 값은 %f\n", g);
16
17     return 0;
18 }
```



산술 연산자 예제2



ROBIT
ROBOT SPORT GAME TEAM

```
03 int main(void)
04 {
05     int a, b, c;
06     a = 10; b = 20; c = 30;
07
08     a = ++b;
09     c = b++;
10     printf("a = %d, b = %d, c = %d\n\n", a, b, c);
11
12
13     a = ++b + ++c;
14     printf("a = ++b + ++c 문장 실행 후\n");
15     printf("a = %d, b = %d, c = %d\n\n", a, b, c);
16
17     a = b++ + c++;
18     printf("a = b++ + c++ 문장 실행 후\n");
19     printf("a = %d, b = %d, c = %d\n", a, b, c);
20
21     return 0;
22 }
```

```
a = 21, b = 22, c = 21

a = ++b + ++c 문장 실행 후
a = 45, b = 23, c = 22

a = b++ + c++ 문장 실행 후
a = 45, b = 24, c = 23
```

대입 연산자



ROBIT
ROBOT SPORT GAME TEAM

변수 = 수식

Ex)

```
x = 10;           // 상수 10을 변수 x에 대입한다.  
y = x;            // 변수 x의 값을 변수 y에 대입한다.  
z = 2*x + 7;      // 수식 2*x + y를 계산하여 변수 z에 대입한다.
```

왼쪽에는 항상 변수가 와야 한다.

Ex)

```
x + 2 = 0;        // Error 왼편이 변수 이름이 아니다.  
2 = x;
```


대입 연산자 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a, b = 0;
06
07     a = 4;
08     printf("a = 4 문장을 실행한 후 a의 값은 %d\n", a);
09
10     a = a + 3;
11     printf("a = a + 3 문장을 실행한 후 a의 값은 %d\n", a);
12
13     b = a + 2;
14     printf("b = a + 2 문장을 실행한 후 b의 값은 %d\n", b);
15
16     return 0;
17 }
```

```
C:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
a = 4 문장을 실행한 후 a의 값은 4
a = a + 3 문장을 실행한 후 a의 값은 7
b = a + 2 문장을 실행한 후 b의 값은 9
```

논리 연산자



RO:BIT
ROBOT SPORT GAME TEAM

- 여러 개의 조건을 조합하여 참과 거짓을 따지는 연산자
- 결과값은 참(1) 아니면 거짓(0)

연산자 기호	의미	사용 예
&&	x와 y가 모두 참이면 참, 그렇지 않으면 거짓	x && y
	x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓	x y
!	x가 참이면 거짓, 거짓이면 참	!x

논리 연산자 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a = 0, b = 1;
06
07     printf("논리 연산 1 && 2의 결과 : %d\\n", 1 && 2);
08     printf("논리 연산 0 && 2의 결과 : %d\\n\\n", 0 && 2);
09
10     printf("논리 연산 0 || 2의 결과 : %d\\n", 0 || 2);
11     printf("논리 연산 0 || 0의 결과 : %d\\n\\n", 0 || 0);
12
13     printf("논리 연산 !a의 결과 : %d\\n", !a);
14     printf("논리 연산 !b의 결과 : %d\\n", !b);
15
16     return 0;
17 }
```

```
"C:\Program Files\Microsoft Visual Studio\MyProjects\Wss\Debug\..."
논리 연산 1 && 2의 결과 : 1
논리 연산 0 && 2의 결과 : 0

논리 연산 0 || 2의 결과 : 1
논리 연산 0 || 0의 결과 : 0

논리 연산 !a의 결과 : 1
논리 연산 !b의 결과 : 0
```



관계 연산자

- 두 개의 피연산자를 비교하는 연산자
- 결과값은 참(1) 아니면 거짓(0)

연산자 기호	의미	사용 예
==	x와 y가 같은가?	x == y
!=	x와 y가 다른가?	x != y
>	x와 y가 큰가?	x > y
<	x와 y가 작은가?	x < y
>=	x와 y가 크거나 같은가?	x >= y
<=	x와 y가 작거나 같은가?	x <= y

관계 연산자 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a = 0, b = 1, c = 1;
06
07     printf("a < b && c > a의 결과 : %d\\n", a < b && c > a);
08     printf("a > b && c > a의 결과 : %d\\n", a > b && c > a);
09     printf("a > b || c > a의 결과 : %d\\n\\n", a > b || c > a);
10
11     printf("괄호를 사용하는 것이 더 좋음\\n");
12     printf("(a > b) || (c > a)의 결과 : %d\\n", (a > b) || (c > a));
13
14     return 0;
15 }
```

```
C:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
a < b && c > a의 결과 : 1
a > b && c > a의 결과 : 0
a > b || c > a의 결과 : 1

괄호를 사용하는 것이 더 좋음
(a > b) || (c > a)의 결과 : 1
```

비트 단위 연산자



ROBIT
ROBOT SPORT GAME TEAM

연산자 기호	의미	사용 예
<<	x의 비트를 y만큼 왼쪽으로 이동	$x \ll y$
>>	x의 비트를 y만큼 오른쪽으로 이동	$x \gg y$
&	x의 비트와 y의 비트를 AND 연산	$x \& y$
	x의 비트와 y의 비트를 OR 연산	$x y$
^	x의 비트와 y의 비트를 XOR 연산	$x \wedge y$
~	x의 비트를 반전	$\sim x$

비트 단위 연산자



ROBIT
ROBOT SPORT GAME TEAM

연산자 기호	연산 예	
<<	0b00001010 << 1	0b00010100
>>	0b00001010 >> 1	0b00000101
&	0b1010 & 0b0011	0b0010
	0b1010 0b0011	0b1011
^	0b1010 ^ 0b0011	0b1001
~	~0b1010	0b0101



sizeof 연산자

- sizeof 연산자
 - 변수나 자료형의 크기를 알고 싶을 때 사용
- sizeof 연산자를 자료형에 사용한 예

```
sizeof(int): 결과 값 4  
sizeof(double): 결과 값 8  
printf("%d\\n", sizeof(int)); : 출력 값 4
```

- sizeof 연산자를 변수에 사용한 예

```
int a; double d;  
sizeof(a): 결과 값 4  
sizeof d: 결과 값 8
```




sizeof 연산자 예제

```
03 int main(void)
04 {
05     char c = 'a';
06     int a = 10;
07     double d = 5.0;
08
09     printf("char 자료형의 크기는 %d바이트다.\n", sizeof(char));
10     printf("int 자료형의 크기는 %d바이트다.\n", sizeof(int));
11     printf("double 자료형의 크기는 %d바이트다.\n\n", sizeof(double));
12
13     printf("변수 c의 크기는 %d바이트다.\n", sizeof(c));
14     printf("변수 a의 크기는 %d바이트다.\n", sizeof(a));
15     printf("변수 d의 크기는 %d바이트다.\n", sizeof(d));
16
17     return 0;
18 }
```

```
C:\Program Files\Microsoft Visual Studio\MyProjects\ss\DebugW...
char 자료형의 크기는 1바이트다.
int 자료형의 크기는 4바이트다.
double 자료형의 크기는 8바이트다.

변수 c의 크기는 1바이트다.
변수 a의 크기는 4바이트다.
변수 d의 크기는 8바이트다.
```

축약 연산자



ROBIT
ROBOT SPORT GAME TEAM

- 축약 연산자
 - 연산 2개를 동시에 수행하여 값을 할당하는 연산자

표 4-11 축약 연산자의 종류와 사용 예

연산자	사용법	설명
$+=$	$a += b$	a와 b를 더하여 a에 대입한다.
$-=$	$a -= b$	a에서 b를 뺀 결과 값을 a에 대입한다.
$*=$	$a *= b$	a와 b를 곱하여 a에 대입한다.
$/=$	$a /= b$	a를 b로 나눈 몫을 a에 대입한다.
$\%=$	$a \% b$	a를 b로 나눈 나머지를 a에 대입한다.
$\ll=$	$a \ll b$	a를 b비트만큼 왼쪽으로 이동하여 a에 대입한다.
$\gg=$	$a \gg b$	a를 b비트만큼 오른쪽으로 이동하여 a에 대입한다.
$\&=$	$a \& b$	a와 b를 비트별로 AND 연산하여 a에 대입한다.
$\ =$	$a \mid b$	a와 b를 비트별로 OR 연산하여 a에 대입한다.
$\^=$	$a \wedge b$	a와 b를 비트별로 XOR 연산하여 a에 대입한다.

축약 연산자 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a = 10;
06     printf("초깃값 : a = 10\n");
07
08     a += 5;
09     printf("a += 5 : %d\n", a);
10
11     a = 10;
12     a -= 5 + 2;
13     printf("a -= 5 + 2 : %d\n", a);
14
15     a = 10;
16     a *= 2 + 5;
17     printf("a *= 2 + 5 : %d\n", a);
18     return 0;
19 }
```

```
C:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
초깃값 : a = 10
a += 5 : 15
a -= 5+2 : 3
a *= 2+5 : 70
```



조건 연산자

- 조건 연산자의 예

```
( x > y ) ? 1 : 2;
```

- x 값이 y 값보다 크면 결과 값이 1이 되고, 그렇지 않으면 결과 값이 2가 됨.

```
max = ( x > y ) ? x : y;
```

- 두 수 중 큰 값을 알아내는 프로그램

```
min = ( x > y ) ? y : x;
```

- 두 수 중 작은 값을 찾는 예

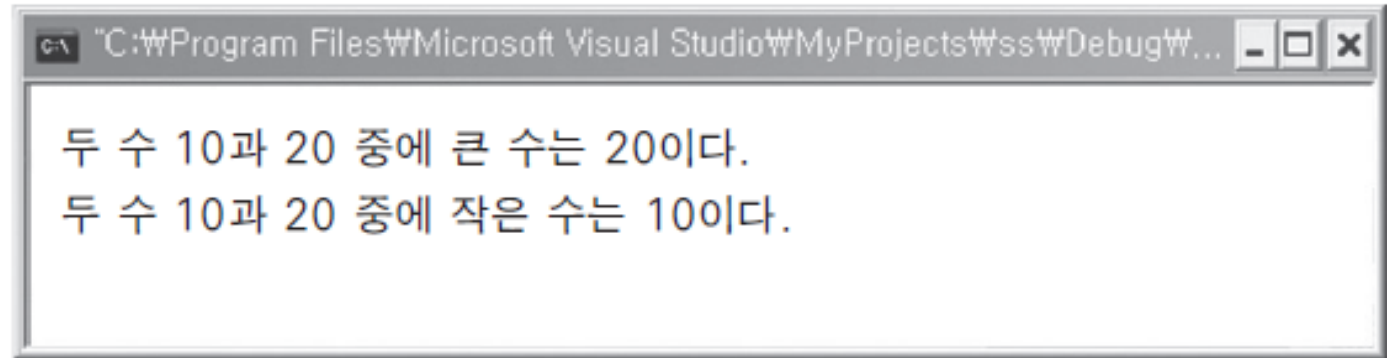
```
value = ( x != 0 ) ? 1 : 0;
```

- x 값이 0이 아니면 value는 1이 되고, x 값이 0이면 value는 0이 됨.



조건 연산자 예제

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int min, max;
06
07     int x = 10, y = 20;
08
09     max = (x > y) ? x : y;
10     min = (x > y) ? y : x;
11
12     printf("두 수 %d과 %d 중에 큰 수는 %d이다.\n", x, y, max);
13     printf("두 수 %d과 %d 중에 작은 수는 %d이다.\n", x, y, min);
14
15     return 0;
16 }
```





ROBIT
ROBOT SPORT GAME TEAM

조건문

조건문



ROBIT
ROBOT SPORT GAME TEAM

1. if 문

if(조건식)

```
{  
    문장;  
}
```

else if(조건식)

```
{  
    문장;  
}
```

else

```
{  
    문장;  
}
```

2. switch 문

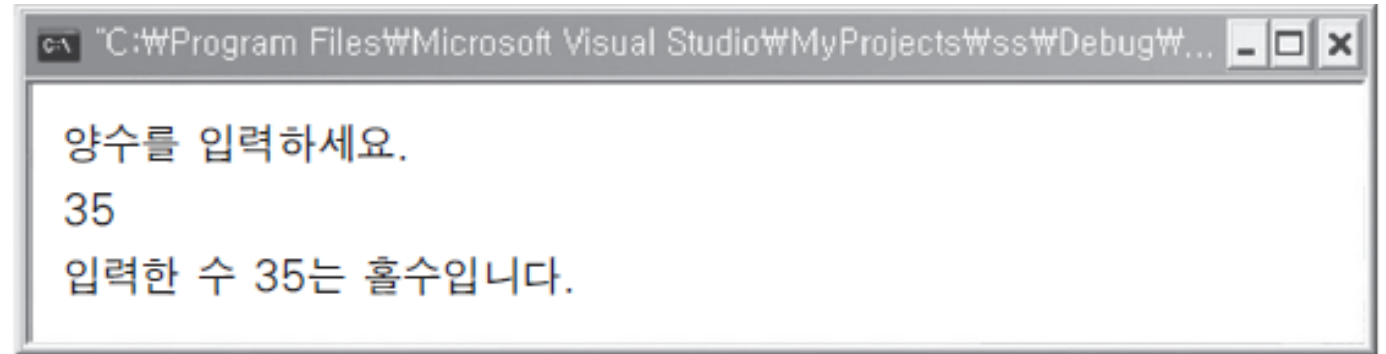
switch(조건식)

```
{  
    case 조건1:  
        문장;  
        break;  
    case 조건2:  
        문장;  
        break;  
    default:  
        문장;  
        break;  
}
```



if문 예제1

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int input;
06     printf("양수를 입력하세요.\n");
07     scanf_s("%d", &input);
08
09     if(input % 2 == 0)
10         printf("입력한 수 %d는 짝수입니다.\n", input);
11     else
12         printf("입력한 수 %d는 홀수입니다.\n", input);
13
14     return 0;
15 }
```



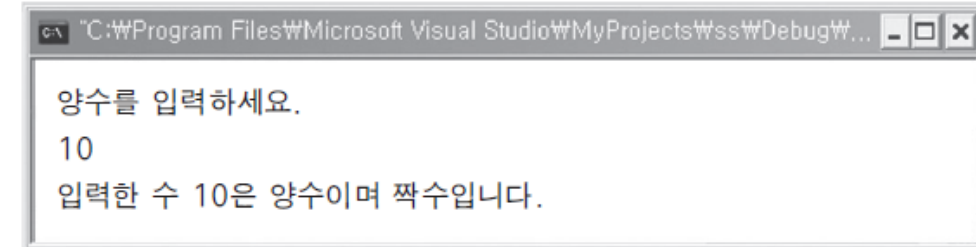
if문 예제2



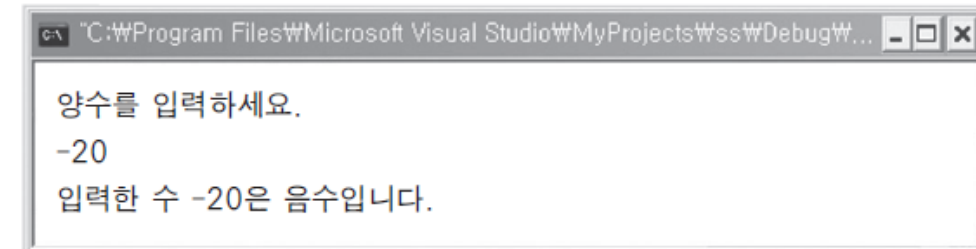
ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int input;
06     printf("양수를 입력하세요.\n");
07     scanf_s("%d", &input);
08
09     if(input >= 0 && input % 2 == 0)
10         printf("입력한 수 %d은 양수이며 짝수입니다.\n", input);
11
12     else if(input >= 0 && input % 2 != 0)
13         printf("입력한 수 %d은 양수이며 홀수입니다.\n", input);
14
15     else
16         printf("입력한 수 %d은 음수입니다.\n", input);
17
18     return 0;
19 }
```

[입력된 수가 양수일 때]



[입력된 수가 음수일 때]



switch~case 예제1



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02 int main(void)
03 {
04     int input;
05
06     printf("영어로 인사하는 법을 배우겠습니다.\n");
07     printf("아침 인사는 1번을 누르세요.\n");
08     printf("점심 인사는 2번을 누르세요.\n");
09     printf("저녁 인사는 3번을 누르세요.\n");
10     scanf_s("%d", &input);
```

```
13     switch(input){
14         case 1 :
15             printf("Good Morning!!\n");
16             break;
17         case 2 :
18             printf("Good Afternoon!!\n");
19             break;
20         case 3 :
21             printf("Good Night!!\n");
22             break;
23         default :
24             printf("잘못 입력했습니다. 1~3 사이의 숫자를 입력하세요.\n");
25     }
```



ROBIT
ROBOT SPORT GAME TEAM

반복문

반복문



ROBIT
ROBOT SPORT GAME TEAM

1. for 문

```
for(초기식;조건식;증감식)
{
    문장;
}
```

2. while 문

```
while(조건식)
{
    문장;
}
```

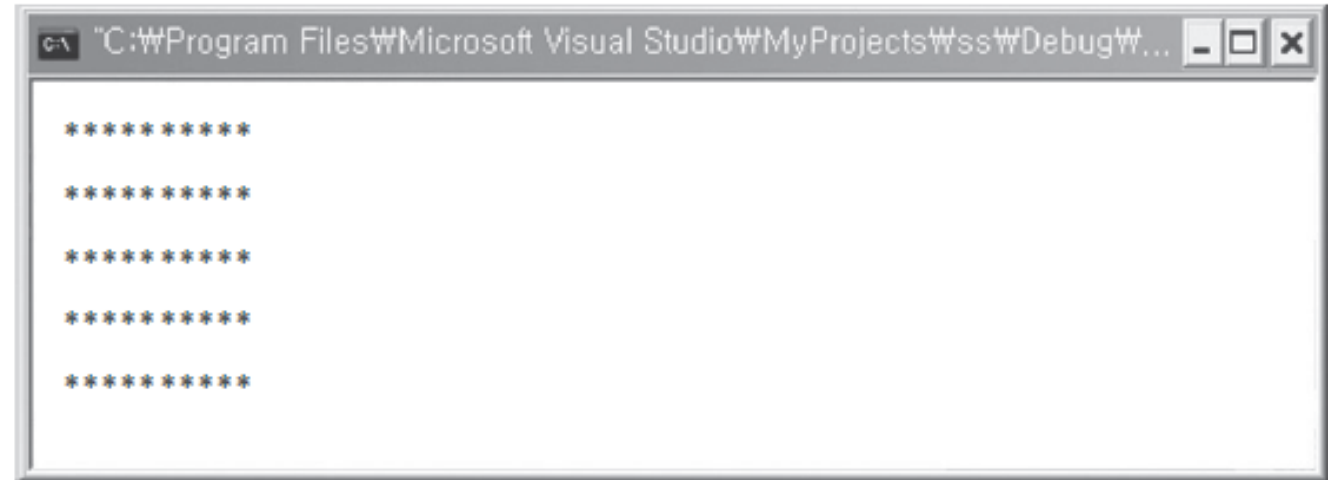
3. do~while 문

```
do
{
    문장;
}while(조건식);
```



for 예제1

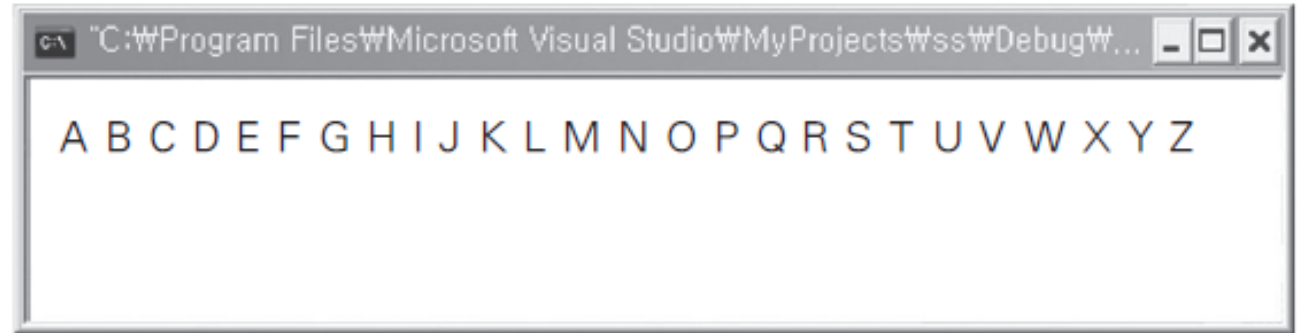
```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06
07     for(i = 0; i < 5; i++)
08         printf("*****\n");    // * 10개
09
10     return 0;
11 }
```





for 예제2

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06     for(i = 65; i <= 90; i++)
07     {
08         printf("%c ", i);
09     }
10     printf("\n");
11     return 0;
12 }
```





for 예제3

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int sum, i, j;
06
07     printf("다중 for문 예제입니다.\n");
08
09     for(i = 1; i <= 3; i++)
10     {
11         for(j = 1; j <= 3; j++)
12         {
13             sum = i + j;
14             printf("%d + %d = %d\n", i, j, sum);
15         }
16     }
17
18     return 0;
19 }
```

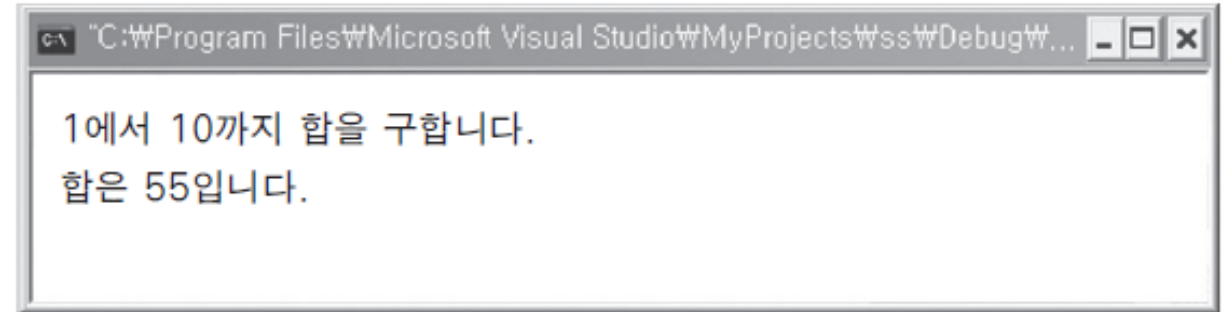
```
C:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
다중 for문 예제입니다.
1 + 1 = 2
1 + 2 = 3
1 + 3 = 4
2 + 1 = 3
2 + 2 = 4
2 + 3 = 5
3 + 1 = 4
3 + 2 = 5
3 + 3 = 6
```

while 예제1



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int sum = 0, number = 1;
06     printf("1에서 10까지 합을 구합니다.\n");
07
08     while(number <= 10)
09     {
10         sum += number;
11         number++;
12     }
13
14     printf("합은 %d입니다.\n", sum);
15     return 0;
16 }
```

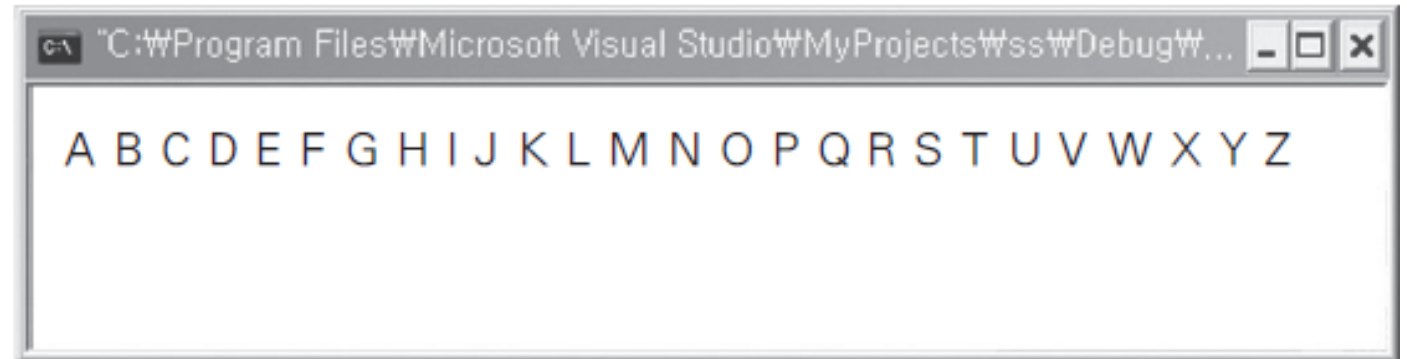


while 예제2



ROBIT
ROBOT SPORT GAME TEAM

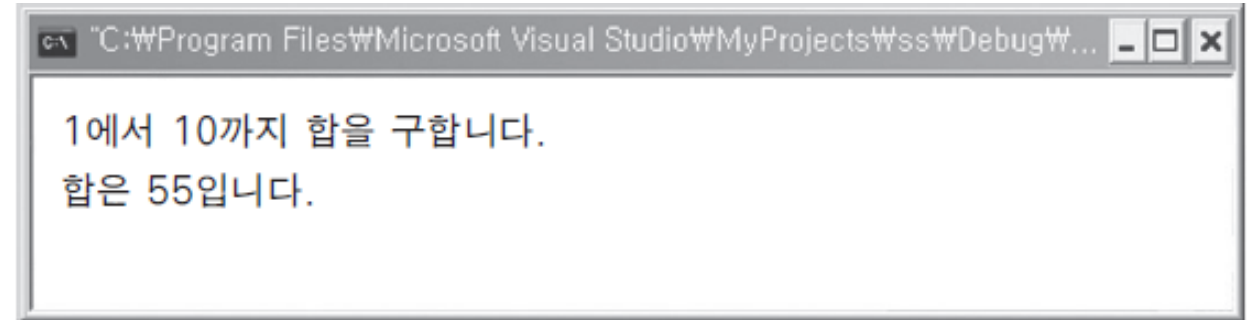
```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i = 65;
06
07     while(i <= 90)
08     {
09         printf("%c ", i);
10         i++;
11     }
12
13     printf("\n");
14     return 0;
15 }
```





do~while 예제1

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int sum = 0;
06     int number = 0;
07
08     printf("1에서 10까지 합을 구합니다.\n");
09
10     do
11     {
12         number++;
13         sum += number;
14     } while(number < 10);
15
16     printf("합은 %d입니다.\n", sum);
17
18     return 0;
19 }
```



무한 반복 하는 방법



ROBIT
ROBOT SPORT GAME TEAM

[for문]

```
for(;;){  
    문장  
    :  
}
```

[while문]

```
while(1){  
    문장  
    :  
}
```



break 와 continue

break문

반복문을 빠져나오거나 switch~case문에서 정상적인 흐름을 변경하기 위해 사용

```
int sum = 0;
for (int i = 1; i <= 10; i++)
{
    if (i % 2 == 0) break;
    sum += i;
    printf("i : %d \n", i);
}
printf("sum : %d", sum);
```

실행결과

```
i : 1
sum : 1
```

continue문

반복문을 빠져나오지 않고 해당 반복문의 다음으로 프로그램의 흐름을 이동

```
int sum = 0;
for (int i = 1; i <= 10; i++)
{
    if (i % 2 == 0) continue;
    sum += i;
    printf("i : %d \n", i);
}
printf("sum : %d", sum);
```

실행결과

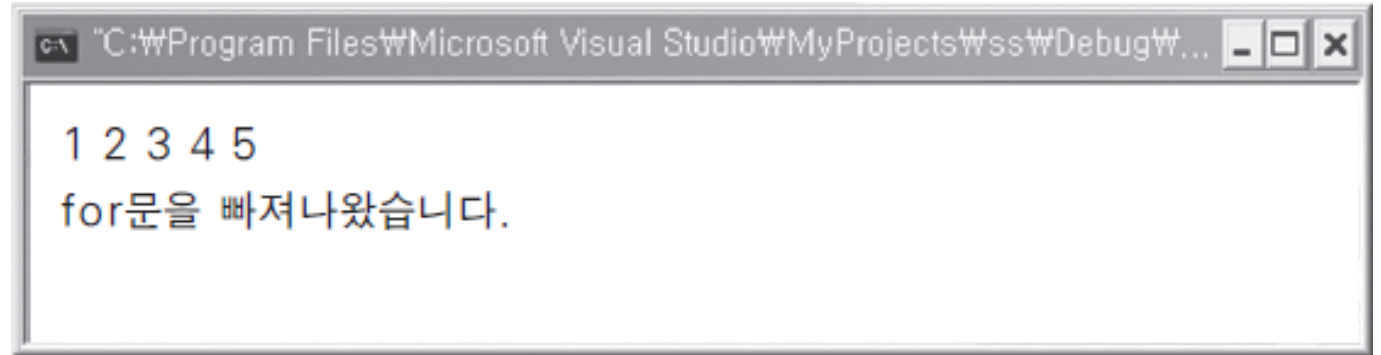
```
i : 1
i : 3
i : 5
i : 7
i : 9
sum : 25
```

break 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06
07     for(i = 1; i < 100; i++)
08     {
09         printf("%d ", i);
10         if(i >= 5) break;
11     }
12
13     printf("\n for문을 빠져나왔습니다.\n");
14
15     return 0;
16 }
```



continue 예제



ROBIT
ROBOT SPORT GAME TEAM

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i, sum = 0;
06
07     for(i = 1; i <= 10; i++)
08     {
09         if((i % 2 == 0) && (i % 3 == 0)) continue;
10         printf("%d ", i);
11     }
12
13     printf("\n");
14     return 0;
15 }
```





ROBIT
ROBOT SPORT GAME TEAM

과제

과제 관련 주의사항



ROBIT
ROBOT SPORT GAME TEAM

1. 제출 기한 : 다음 날 수업 전까지
2. 제출 형식 : 프로젝트 파일, 보고서(PDF) 를 (로빗_18기_수습단원_이름) 으로 압축 후
kwrobit2023@gmail.com 으로 제출
(보고서는 코드 설명과 실행 화면 첨부, 메일 제목은 C언어_n일차_이름)
3. 채점 기준 :
 - 1) 프로그램의 실행가능 여부
 - 2) 교육하지 않은 C언어 개념 사용 시 감점
 - 3) 예외처리
 - 4) 효율적인 코드 작성
 - 5) 제출 형식

과제 1



ROBIT
ROBOT SPORT GAME TEAM

5개의 실수를 입력받아 평균과 최댓값 최솟값을 출력하는 프로그램을 작성하시오.

```
1 번째 실수를 입력하시오. 1
2 번째 실수를 입력하시오. 1.5
3 번째 실수를 입력하시오. 2
4 번째 실수를 입력하시오. 2.5
5 번째 실수를 입력하시오. 3
---- 결과 ----
평균은 2.000000입니다.
최댓값은 3.000000입니다.
최솟값은 1.000000입니다.
```

과제2



ROBIT
ROBOT SPORT GAME TEAM

피보나치 수는 0과 1로 시작한다. 1번째 피보나치 수는 0이고, 2번째 피보나치 수는 1이다.

그 다음 3번째 부터는 바로 앞 두 피보나치 수의 합이 된다. 이를 식으로 써보면 $F_n = F_{n-1} + F_{n-2}$ ($n \geq 2$)가 된다.

피보나치 수를 나열하여 작성하면 아래와 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987...

n을 입력받아, n번째 피보나치 수를 출력하는 프로그램을 작성하시오.

과제3

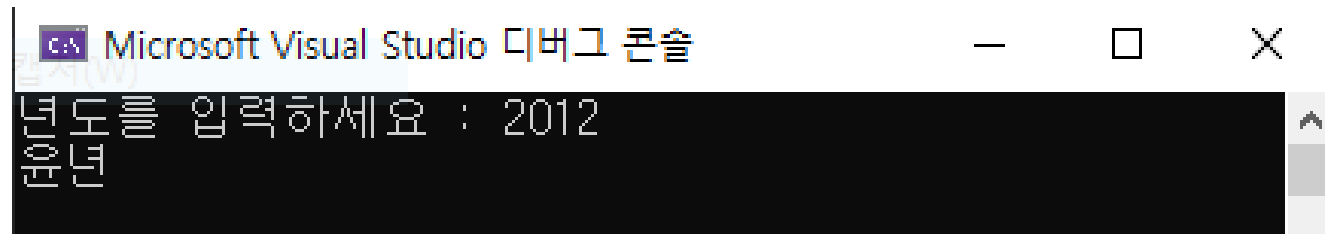


ROBIT
ROBOT SPORT GAME TEAM

연도가 주어졌을 때, 윤년인지 판단하는 프로그램을 작성하시오.

윤년은 연도가 4의 배수이면서, 100의 배수가 아닐 때 또는 400의 배수일 때이다.

예를 들어, 2012년은 4의 배수이면서 100의 배수가 아니 라서 윤년이다. 1900년은 100의 배수이고 400의 배수는 아니기 때문에 윤년이 아니다. 하지만, 2000년은 400의 배수이기 때문에 윤년이다.



과제4



ROBIT
ROBOT SPORT GAME TEAM

Input Arithmetic Operation

ex) $3.4 * 8.5$
 $2.9 - 5.4$
 $3.9 * 8.0$
 $3.9 ^ 8$

input : $3 + 5$

$3.00 + 5.00 = 8.00$

계속하려면 아무 키나 누르십시오 . . .

Input Arithmetic Operation

ex) $3.4 * 8.5$
 $2.9 - 5.4$
 $3.9 * 8.0$
 $3.9 ^ 8$

input : $5.57 ^ 4$

$5.57 ^ 4.00 = 962.54$

계속하려면 아무 키나 누르십시오 . . .

Input Arithmetic Operation

ex) $3.4 * 8.5$
 $2.9 - 5.4$
 $3.9 * 8.0$
 $3.9 ^ 8$

input : $5 / 2.5$

$5.00 / 2.50 = 2.00$

계속하려면 아무 키나 누르십시오 . . .

Input Arithmetic Operation

ex) $3.4 * 8.5$
 $2.9 - 5.4$
 $3.9 * 8.0$
 $3.9 ^ 8$

input : $8.6 - 3$

$8.60 - 3.00 = 5.60$

계속하려면 아무 키나 누르십시오 . . .

Input Arithmetic Operation

ex) $3.4 * 8.5$
 $2.9 - 5.4$
 $3.9 * 8.0$
 $3.9 ^ 8$

input : $5.0 * 5.78$

$5.00 * 5.78 = 28.90$

계속하려면 아무 키나 누르십시오 . . .

사칙 연산, 거듭제곱이 가능한 계산기 프로그램을 작성하시오.

과제5



RO:BIT
ROBOT SPORT GAME TEAM

아래 예제를 통해 규칙을 유추하여, 값을 입력 받은 후 별을 출력하는 프로그램을 작성하시오.

값을 입력하세요.1
**

값을 입력하세요.2
* *

* *

값을 입력하세요.3
* *
** **

** **
* *

값을 입력하세요.4
* *
** **
*** ***

*** ***
** **
* *

과제6



ROBIT
ROBOT SPORT GAME TEAM

아래 예제를 통해 규칙을 유추하여, 값을 입력 받은 후 별을 출력하는 프로그램을 작성하시오.

값을 입력하세요. 1

*

값을 입력하세요. 2

*

값을 입력하세요. 3

*

* *

값을 입력하세요. 4

*

* *

* *

과제7



ROBIT
ROBOT SPORT GAME TEAM

n과 r을 입력 받아
순열, 중복 순열, 조합, 중복조합을
구하는 코드를 작성하시오.

순열 ${}_nP_r = \frac{n!}{(n-r)!}$

중복순열 ${}_n\Pi_r = n^r$

조합 ${}_nC_r = \frac{n!}{(n-r)!r!}$

중복조합 ${}_nH_r = {}_{n+r-1}C_r$