

2장 상수와 데이터 출력



❖ 프로그램과 main 함수 구조

- main 함수 - 프로그램이 시작되는 곳으로 필수 함수
- 머리(head)와 몸통(body)로 구성

10과 20을 더하는 프로그램

소스 코드 예제2-1.c

```
01  /*  작성자 : 홍길동
02      제목 : 10과 20을 더하는 프로그램 */
03
04  int main(void)
05  {
06      10 + 20;          // 10과 20을 더한다.
07
08      return 0;         // 프로그램 종료
09  }
```

머리

```
int main(void)
```

몸통

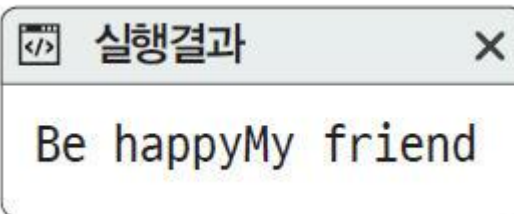
```
{
    // 실행코드
    return 0;
}
```

❖ 출력 함수(printf)의 사용법 : 문자열 출력

문자열을 화면에 출력하는 프로그램

소스 코드 예제2-2.c

```
01  #include <stdio.h>           // stdio : standard input
02                                   // output(표준입출력)의 줄임말
03  int main(void)
04  {
05      printf("Be happy");        // 문자열 "Be happy" 출력
06      printf("My friend");       // 문자열 "My friend" 출력
07
08      return 0;
09  }
```



실행결과

Be happyMy friend

❖ 제어 문자 출력

제어 문자를 사용한 출력

소스 코드 예제2-3.c

```

01 #include <stdio.h>
02
03 int main(void)
04 {
05     printf("Be happy\n");           // "Be happy"를 출력하고 줄을 바꿈(\n)
06     printf("12345678901234567890\n"); // 화면에 열 번호 출력하고 줄을 바꿈(\n)
07     printf("My\tfriend\n");
    // "My"를 출력하고 탭 위치로 이동(\t) 후에 "friend"를 출력하고 줄을 바꿈(\n)
08     printf("Goot\bd\tchance\n");    // "Goot"를 출력하고 한 칸 왼쪽으로 이동(\b)해
    // t를 d로 바꾸고 탭 위치로 이동(\t) 후에 "chance"를 출력하고 줄을 바꿈(\n)
09     printf("Cow\rW\a\n");
    // 맨 앞으로 이동(\r)해 C를 W로 바꾸고 벨소리(\a)를 내고 줄을 바꿈(\n)
10
11     return 0;
12 }

```

실행결과

```

Be happy
12345678901234567890
My      friend
Good    chance
Wow
이 창을 닫으려면 아무 키나 누르세요.

```

❖ 정수와 실수 출력

정수와 실수의 출력

소스 코드 예제2-4.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     printf("%d\n", 10);           // %d 위치에 10 출력
06     printf("%lf\n", 3.4);        // %lf 위치에 3.4를 소수점 이하 6자리까지 출력
07     printf("%.1lf\n", 3.45);     // 소수점 이하 첫째 자리까지 출력(둘째 자리에서 반올림)
08     printf("%.10lf\n", 3.4);     // 소수점 이하 10자리까지 출력
09
10     printf("%d과 %d의 합은 %d입니다.\n", 10, 20, 10 + 20);
11     printf("%.1lf-%.1lf = %.1lf\n", 3.4, 1.2, 3.4 - 1.2);
12
13     return 0;
14 }
```

실행결과

```
10
3.400000
3.5
3.4000000000
10과 20의 합은 30입니다.
3.4-1.2 = 2.2
```

키워드로 끝내는 핵심 포인트

- ❖ C 프로그램은 **main 함수**로 시작한다.
- ❖ `//`는 한 줄 **주석문**, `/* */`는 여러 줄 주석문이다.
- ❖ **printf 함수**는 데이터를 화면에 출력한다.
- ❖ **제어 문자**의 기능에 따라 문자열의 출력 형태가 바뀐다.
- ❖ printf 함수로 정수 출력은 `%d`, 실수는 `%lf` **변환 문자** 사용

표로 정리하는 핵심 포인트

표 2-1 printf 함수의 사용법

출력 데이터	사용 예	출력 결과
문자열	<code>printf("Be happy!");</code>	Be happy! 출력
제어 문자	<code>printf("Be happy!\n");</code>	Be happy! 출력 후 줄 바꿈
정수	<code>printf("%d", 10);</code>	정수 10 출력(변환 문자 %d)
실수	<code>printf("%lf", 3.5);</code>	실수 3.500000 출력(변환 문자 %lf)
수식	<code>printf("%d", 10 + 20);</code>	10과 20을 더한 결과인 30 출력

표 2-2 제어 문자의 종류

제어 문자	의미	기능
<code>\n</code>	개행(new line)	줄을 바꾼다.
<code>\t</code>	탭(tab)	출력 위치를 다음 탭(tab) 위치로 옮긴다.
<code>\r</code>	캐리지 리턴(carriage return)	출력 위치를 줄의 맨 앞으로 옮긴다.
<code>\b</code>	백스페이스(backspace)	출력 위치를 한 칸 왼쪽으로 옮긴다.
<code>\a</code>	알럿(alert) 경보	벨(bell)소리를 낸다.

❖ 정수 상수 표현법 (1/2)

■ 진법별 수 표현 방법

10진수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
8진수	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
16진수	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

①: 10진수 10 → 8진수 10
②: 8진수 10 → 16진수 A

■ C 언어에서 진법 표현하기

10진수

12

8진수

014

16진수

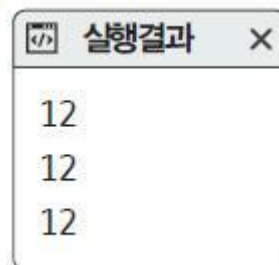
0xc

❖ 정수 상수 표현법 (2/2)

세 가지 진법의 정수 상수

소스 코드 예제2-5.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     printf("%d\n", 12);           // 10진수 정수 상수 출력
06     printf("%d\n", 014);         // 8진수 정수 상수 출력
07     printf("%d\n", 0xc);         // 16진수 정수 상수 출력
08
09     return 0;
10 }
```

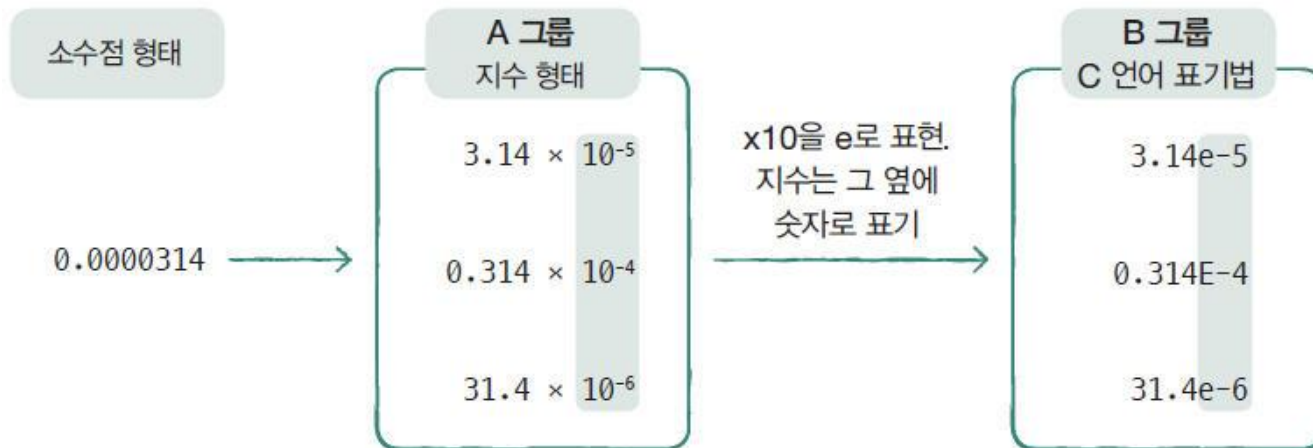


❖ 실수 상수 표현법 (1/3)

■ 소수점 표기법

3.4 -1.5 +10.0
 .5 -10. ← 소수점 앞이나 뒤의 무의미한 0은 생략할 수 있습니다.

■ 지수 표기법



❖ 실수 상수 표현법 (2/3)

- 정규화(normalization) 표기법
 - 소수점 앞에 0이 아닌 한 자리만을 사용한 지수

소수점 형태 같은 값을 표현한 다양한 지수 형태

0.0000314 3.14e-5 0.0314e-3 0.00000314e1 314e-7

정규화

❖ 실수 상수 표현법 (3/3)

지수 형태의 실수 상수

소스 코드 예제2-6.c

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     printf("%.1lf\n", 1e6);           // 지수 형태의 실수를 소수점 형태로 출력
06     printf("%.7lf\n", 3.14e-5);       // 소수점 이하 7자리까지 출력
07     printf("%le\n", 0.0000314);       // 소수점 형태의 실수를 지수 형태로 출력
08     printf("%.2le\n", 0.0000314);     // 지수 형태로 소수점 이하 둘째 자리까지 출력
09
10     return 0;
11 }
```

실행결과
1000000.0
0.0000314
3.140000e-05
3.14e-05

❖ 문자와 문자열 상수 표현법

- 문자는 작은 따옴표, 문자열은 큰 따옴표로 묶는다.

문자와 문자열 데이터의 출력

소스 코드 예제2-7.c

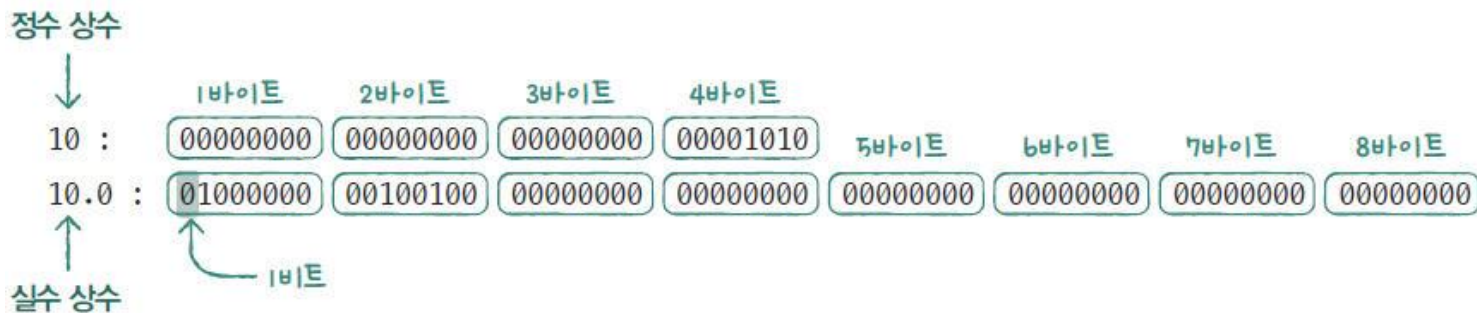
```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     printf("%c\n", 'A');           // 문자 상수 출력
06     printf("%s\n", "A");          // 문자열 상수 출력
07     printf("%c은 %s입니다.\n", '1', "first"); // 문자(%c)와 문자열(%s)을 함께 출력
08
09     return 0;
10 }
```

실행결과

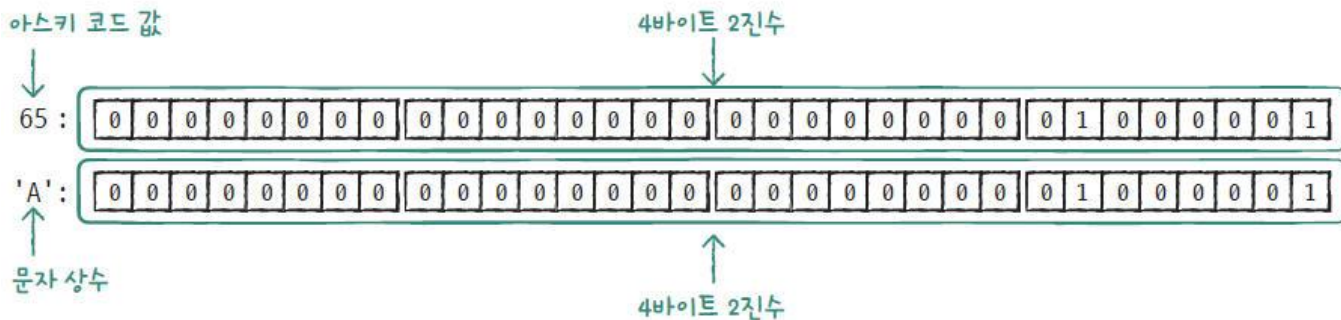
A
A
1은 first입니다.

❖ 상수가 컴파일된 후의 데이터 크기

- 정수는 4바이트, 실수는 8바이트의 크기로 변환

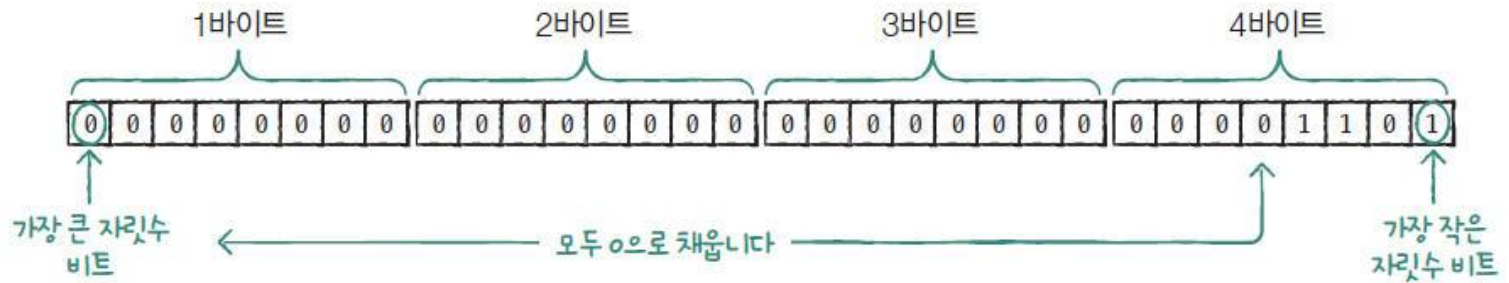


- 문자 상수는 아스키코드값 정수와 같은 형태로 변환



❖ 정수 상수가 컴파일된 후의 비트 형태 (1/2)

- 4바이트 크기의 2진수로 변환(양수 13의 예)

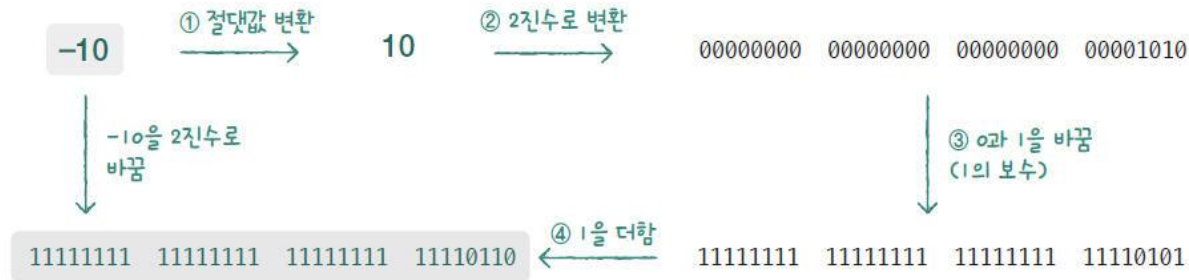


- 비트 수에 따라 표현할 수 있는 값의 범위

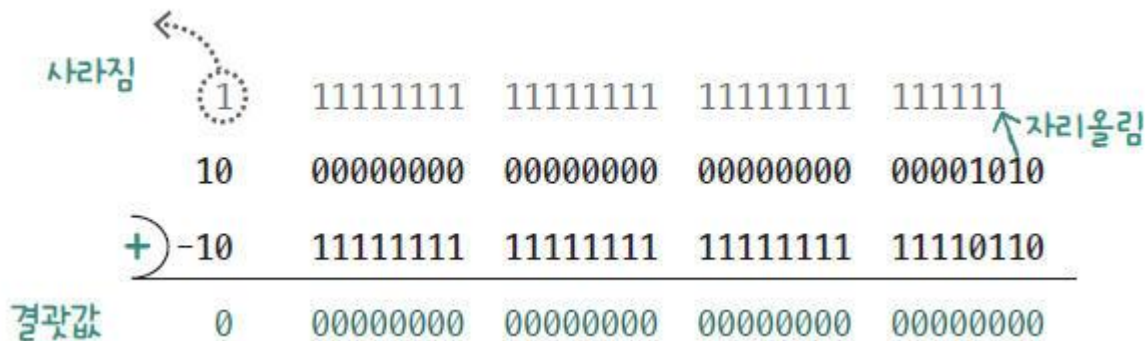
비트 수	표현할 수 있는 값의 범위	
1비트	$0 \sim 2^1 - 1$	$0 \sim 1$
8비트	$0 \sim 2^8 - 1$	$0 \sim 255$
32비트	$0 \sim 2^{32} - 1$	$0 \sim 4294967295$

❖ 정수 상수가 컴파일된 후의 비트 형태 (2/2)

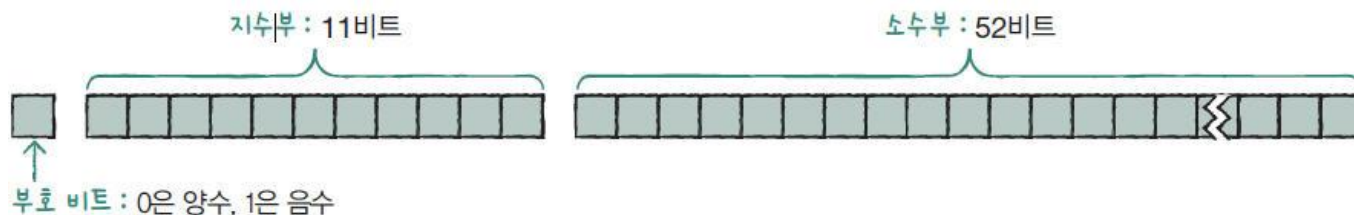
- 음수는 절댓값을 2의 보수로 바꾸어 변환(음수 -10의 예)



- 2의 보수로 변환시 양수와 음수를 쉽게 더할 수 있다.



- IEEE 754 표준의 double 형식에 따라 변환



- **값의 표현 범위**

$$-1.79 \times 10^{308} \sim 1.79 \times 10^{308}$$

키워드로 끝내는 핵심 포인트

- ❖ 10은 정수 상수, 10.0은 실수 상수
- ❖ 'A'는 문자 상수, "A" 문자열 상수
- ❖ 양수는 4바이트 크기의 2진수로, 음수는 2의 보수로 컴파일
- ❖ 실수는 IEEE 754 표준의 double 형식에 따라 번역
- ❖ 첫 비트는 부호 비트, 이후 11개 비트는 지수부, 나머지 52비트는 소수부를 뜻한다.

표로 정리하는 핵심 포인트

표 2-3 상수 종류

종류	표현 방법	사용 예
정수	0~9, +, - 기호 사용	10, -5, +20, 0
실수	0~9, +, -, .(소수점) 기호 사용	3.4, -1.7, .5, 10.0
문자	문자를 작은따옴표로 묶음	'A', 'b', '0', '*',
문자열	하나 이상의 문자를 큰따옴표로 묶음	"A", "banana"

표 2-4 진법별 수의 표현 방법

컴퓨터에서 수를 다룰 때는 대부분 0으로 시작한다는 점을 꼭 기억하세요.

10진수	8진수	16진수	2진수	10진수	8진수	16진수	2진수
0	00	0x0	0	8	010	0x8	1000
1	01	0x1	1	9	011	0x9	1001
2	02	0x2	10	10	012	0xa	1010
3	03	0x3	11	11	013	0xb	1011
4	04	0x4	100	12	014	0xc	1100
5	05	0x5	101	13	015	0xd	1101
6	06	0x6	110	14	016	0xe	1110
7	07	0x7	111	15	017	0xf	1111