Fast campus

CS SCHOOL

자료형

정수를 나타내는 자료형

1. 정수

1) int
4 byte, 표현 범위 : -2,147,483,648 ~ 2,147,483,647

2) Short (거의 쓰이지 않고 소켓 프로그래밍에서만 쓰임)
2 byte, 표현 범위 : -32,768 ~ 32,767

3) char (문자 표현을 위해 탄생한 자료형, 하지만 어쨌든 정수 표현 가능)
1 byte, 표현 범위 : -128 ~ 127

자료형

실수를 나타내는 자료형

2. 실수 (표현 범위는 넓지만 정확도가 떨어집니다)

float? double? 선택 기준은?

메모리가 엄청난 이슈가 아니라면 무조건 double!!

따져야 한다면 가장 중요하게 볼 것은 정밀도!!

자료형	표현범위	소수점 이하 정밀도	바이트 수
float	10^{-37} 이상 10^{38} 이하	6자리	4
double	10^{-307} 이상 10^{308} 이하	15자리	8

Fast campus

자료형(Data type)

구분	자료형	범위	바이트
정수형	char unsigned char	-128 ~ 127 0 ~ 255	1(8) 1(8)
	short int long unsigned short unsigned int unsigned long	-32768 ~ 32767 -2,147,483,648 ~ 2,147,483,647 -2,147,483,648 ~ 2,147,483,647 0~65535 0~4,294,967,295 0~4,294,967,295	2(16) 4(32) 4(32) 2(16) 4(32) 4(32)
실수형	float double	$8.4X10^{-37}$ ~ $3.4X10^{38}$ $2.2X10^{-308}$ ~ $1.8X10^{308}$	4(32) 8(64)
나열형	enum	정수를 대신하여 사용하는 별명, int형의 크기	
무치형	void	실제 자료는 없음을 명시적으로 선언	

Fast campus

2진수와 16진수

1. 10진수
   : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
2. 2진수
   : 0, 1
3. 16진수
   : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

- 진수 변환

1. 10진수 -> 2진수
   예) 25 -> 16 + 8 + 1 = $1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$
   = 11001

2. 2진수 -> 16진수 (2진수 4자리가 16진수 1자리 수)
   예) 1001 -> 8 + 1 = 9
   1010 -> 8 + 2 = 10 = a
   1111 -> 8 + 4+ 2+1 = 15 = f

2의 보수(two's complement)

: 컴퓨터가 수를 저장하는 방식1
- 정수형

1. 2의 보수란?

```
  101011
+ 010100  (101011의 1의 보수)
--------
  111111  (자리수가 바뀌기 전의 수)
```

2의 보수 만드는 방법
: 1의 보수 +1

010101 (101011의 2의 보수)

Fast campus

2의 보수(two's complement)

: 컴퓨터가 수를 저장하는 방식1
- 정수형

2. 정수를 저장하는 방식

char형의 경우

1) 양의 정수

0	000 1010
sign	number

10진수 : 10

그렇다면 음의 정수 -10은

1 000 1010인가??

정답은 NO!

2) 음의 정수

1	111 0110
sign	2의 보수

111 0110

1의 보수

000 1001

+1

2의 보수

000 1010

10진수 : -10

부동소수점(floating point number)

: 컴퓨터가 수를 저장하는 방식2

- 실수형

3. 실수를 저장하는 방식

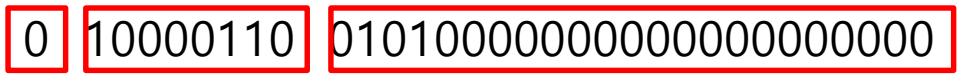
$$\pm 1.m \times 2^{e-b}$$

m : mantissa(가수)
e : exponent(지수)
b : bias(바이어스수)

$$Bias = 2^{n-1} - 1$$

float의 경우 127

float형의 경우



0x43280000(16진수)

sign exponent mantissa

$1.0101 \times 2^{e-b}$

$e = 2^7 + 2^2 + 2^1 = 134$

b = 127

$1.0101 \times 2^7 = 1 * 2^7 + 1*2^{7-2} + 2^{7-4} = 168.0$

Fast campus

부동소수점(floating point number)

: 컴퓨터가 수를 저장하는 방식2

- 실수형

3. 실수를 저장하는 방식

$$\pm 1.m \times 2^{e-b}$$

m : mantissa(가수)
e : exponent(지수)
b : bias(바이어스수)

float형의 경우

$$\text{Bias} = 2^{n-1} - 1$$

float의 경우 127

17.25를 float형으로 나타내보자.

e-b = 4
e-127 = 4
e = 131

1) 2진수로

$17.25 = 16 + 1 + 0.25 = 2^4 + 2^0 + 2^{-2} = 10001.01$

2) 정규화 : 가수의 첫 번째 수를 밑보다 작은 자연수로

$10001.01 = 1.000101 \times 2^4$

0 | 10000011 | 00010100000000000000000

0x418a0000 (16진수)

부동소수점(floating point number)

: 컴퓨터가 수를 저장하는 방식2

- 실수형

3. Double 형일 경우

$$\pm 1.m \times 2^{e-b}$$

m : mantissa(가수)
e : exponent(지수)
b : bias(바이어스수)

- sing : 1 bit
- exponent : 11 bit
- mantissa : 52 bit

$$\text{Bias} = 2^{11-1} - 1 = 1023$$

double의 경우 1023

부동소수점(floating point number)
: 컴퓨터가 수를 저장하는 방식2
- 실수형

4. 2의 지수를 쉽게 알 수 있는 꿀팁

0.511718750은 2진수로??

$\log_2 0.511718750 = -0.966576998$

-0.966576998 > -1,
which means 2^{-1} 을 포함

0.511718750 = 0.5 + 0.011718750

$\log_2 0.011718750 = -6.415037499$

-6.415037499 > -7,
which means 2^{-7} 을 포함

0.511718750 = 0.5 + 0.007812500 + 0.003906250

$\log_2 0.003906250 = -8$

-8 = -8,
which means 2^{-8} 이다

0.10000011