C언어 기초 part. 1

Week 2 – 데이터표현, 연산자, 입출력

QnA 메일 : edujongkook@gmail.com

Pdf 파일 : github.com/edujongkook
/pdf_sbs_c_weekend

목차 A table of contents

복습 C언어의 기본 자료형 p-35

문자	알파벳 문자 및 기호 : 'A' , '+' , '&' …	char
정수	양의 정수, 0, 음의 정수 : 10, 100, -50 …	short int long long long
실수	소수점 이하 값을 가진 숫자 : 3.14, 0.01, -5.5 …	float double long double

복습 정수, 실수의 printf 출력

```c
#include <stdio.h>

int main(void) {
    int number = 10;
    float pi = 3.14;
    printf("number : %d₩n", number);
    printf("pi : %f₩n", pi);
    return 0;
}
```

실수형태의 자료형 float을

출력 할때는 %f 를 사용합니다.

복습 문자와 문자열의 printf 출력

```c
#include <stdio.h>

int main(void) {
    char ch = 'A';
    char str[] = "abc";
    printf("ch : %c₩n", ch);
    printf("str : %s₩n", str);
}
```

%c 는 character(문자) 의 약자입니다.

%s 는 string(문자열) 의 약자입니다.

C에서 문자열은 char 자료형 변수를 배열 또는 포인터를 사용하여 표현합니다.

str[ ] 은 배열입니다. (추후 다시 다룹니다)

변수를 사용한 표현 예시

```c
#include <stdio.h>

int main(void) {
    char name[] = "꼬실";
    char type[] = "고양이";
    int age = 4;
    float weight = 4.5;

    printf("이름: %s₩n",name);
    printf("종류: %s₩n",type);
    printf("나이: %d살₩n", age);
    printf("몸무게: %fkg", weight);
}
```

1.

상수와 형변환

1 상수에 대한 이해 p-39

```c
int main(void) {
    const float PI = 3.14;
    // 상수로 저장된 pi
}
```

상수와 변수의 차이

상수와 변수 모두 값을 저장한다는 공통점이 있지만

변수는 값을 변경할 수 있는 반면,

상수는 한번 초기화 된 값을 프로그램 종료 시까지

변경할 수 없음

1 상수에 대한 이해 p-39

1 리터럴 상수

```c
#include <stdio.h>

int main(void) {
    printf("%d", 100);
}
```

100 과 같은 수 그 자체를 리터럴(literal) 상수라고 합니다.

변수와 마찬가지로 메모리에 저장되며 고정되어 바꿀 수 없습니다.

1 상수에 대한 이해 p-39

2 매크로 상수

```c
#include <stdio.h>
#define PI 3.14

int main(void) {
    printf("%f", PI);
}
```

main() 함수 바깥에서 #define 으로 정의되는 수를 매크로(macro) 상수라고 합니다.

매크로 영역의 코드는 전처리 과정에서 별도로 처리되어 코드에 포함됩니다.

1 상수에 대한 이해 p-39

3 심볼릭 상수

```c
#include <stdio.h>

int main(void) {
    const float PI = 3.14;
    printf("%f", PI);
}
```

변수처럼 생성하는 상수를 심볼릭(symbolic) 상수라고 합니다. 변수와 비슷한 영역에 저장되고 값이 초기화되면 변경할 수 없습니다.

1 상수에 대한 이해 p-39

다양한 상수 사용 예

```c
#include <stdio.h>

int main(void) {
    const float PI = 3.14;
    const int MAX_SPEED = 60;
    const int DAY_OF_YEAR = 365;
    const int MONTH_OF_YEAR = 12;
}
```

1 상수 값을 변경하려고 할 때 p-41

```c
#include <stdio.h>
#define LENGTH 10

int main(void) {
    int number = 3;
    const int NUMBER = 5;
    number = 10;
    NUMBER = 10;

    printf("%d₩n", LENGTH);
    printf("%d₩n", number);
    printf("%d₩n", NUMBER);
}
```

1 리터럴 상수와 접미사 p-39

리터럴 상수

접미사	자료형	사용 예
L	long	1234L
LL	long long	34123LL
F	float	3.141F
L	long double	3.541L

```c
#include <stdio.h>

int main(void) {
    printf("%d₩n", sizeof(100));
    printf("%d₩n", sizeof(100LL));
}
```

1 자료형의 변환 p-42

1 자동 형 변환

```
int main(void) {
    double num1 = 10;
    int num2 = 1.2345;
    short num3 = 70000;

    printf("%f₩n", num1);
    printf("%d₩n", num2);
    printf("%d₩n", num3);
    return 0;
}
```

1 자료형의 변환 p-42

2 명시적 형 변환

```c
#include <stdio.h>

int main(void) {
    printf("%d₩n", (short)3.1415);
    printf("%d₩n", (int)3.1415);
    printf("%f₩n", (double)10);
    printf("%f₩n", (float)10);
    return 0;
}
```

2.

데이터의 bit 표현방식

2 C언어의 기본 자료형 p-35

자료형	형태	크기	값의 범위
char	정수(문자)	1 바이트	-128 ~ 127
short	정수	2 바이트	-32,768 ~ 32,767
int		4 바이트	-2,147,483,648 ~ 2,147,483,647
long		4 바이트	-2,147,483,648 ~ 2,147,483,647
long long		8 바이트	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
float	실수	4 바이트	1.175494e-38 ~ 3.402823e+38
double		8 바이트	2.225074e-308 ~ 1.797693e+308
long double		8 바이트 이상	2.225074e-308 ~ 1.797693e+308

2 문자와 아스키코드 p-38

```c
int main(void) {
    char ch = 'A';

    printf("문자로 출력 : %c ₩n", ch);
    printf("숫자로 출력 : %d ₩n", ch);
}
```

'A' 와 같은 문자는 각각 고유한 숫자로 변환되어 처리됩니다.

2 문자와 아스키코드 p-38

```c
int main(void) {
    int num = 66;

    printf("문자로 출력 : %c ₩n", num);
    printf("숫자로 출력 : %d ₩n", num);
}
```

아스키 코드에서 숫자와 매칭되는 문자로 변환하여 출력됩니다.

'B' <-> 66

2 문자와 아스키코드 p-38

```
int main(void) {
    char ch1 = 66, ch2 = 'B';
    short sh1 = 67;
    int in1 = 68;

    printf("%c₩n", ch1);
    printf("%c₩n", ch2);
    printf("%c₩n", sh1);
    printf("%c₩n", in1);
    return 0;
}
```

065	A	089	Y	113	q
066	B	090	Z	114	r
067	C	091	[	115	s
068	D	092	\	116	t
069	E	093	]	117	u
070	F	094	^	118	v
071	G	095	_	119	w
072	H	096	`	120	x
073	I	097	a	121	y
074	J	098	b	122	z
075	K	099	c	123	{
076	L	100	d	124	|
077	M	101	e	125	}
078	N	102	f	126	~
079	O	103	g		

아스키 코드 일부분

전체 아스키코드 p-345

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

문자와 아스키코드 p-38

문자

알파벳 문자 및 기호 : 'A', '+', '&' …

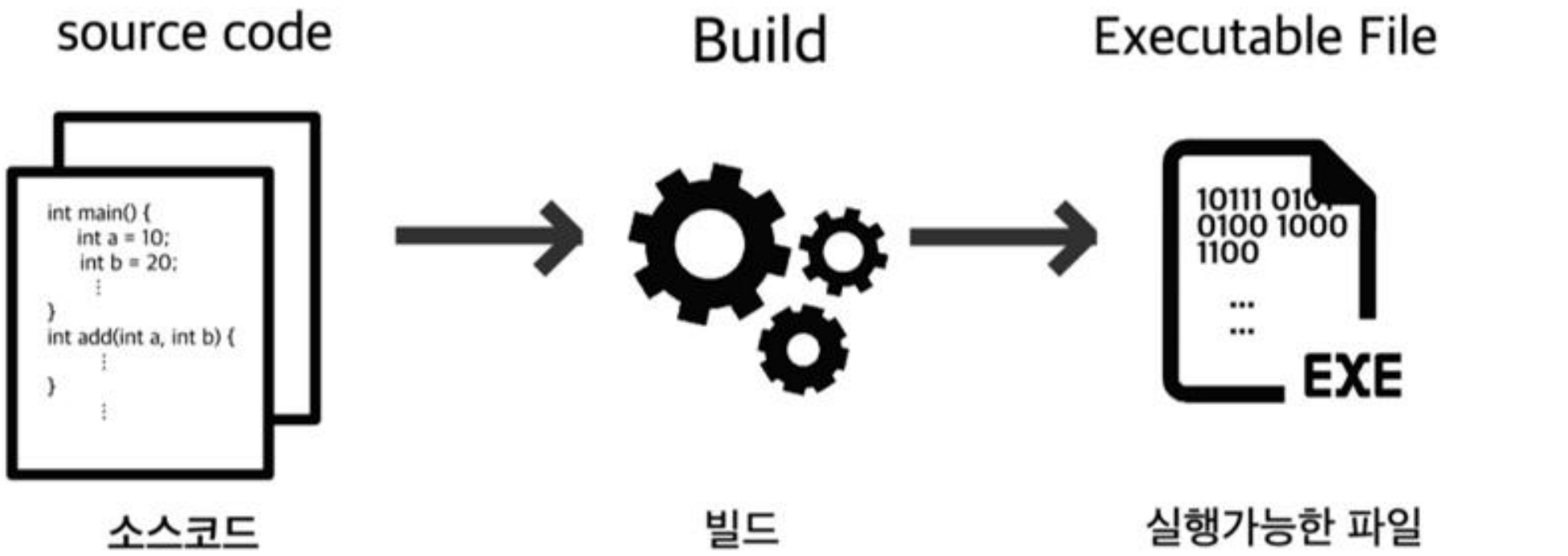
char

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

아스키코드

정수로 저장

2 2진수의 이해 p-48

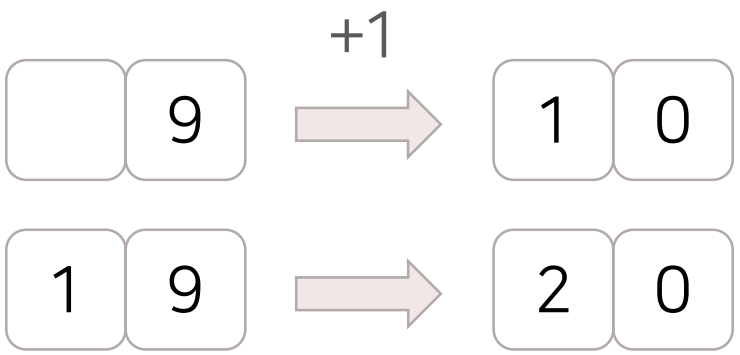


11010010010000010000000111111000000000001111011110000000000111111011001111111100000000

2 2진수의 이해 p-48

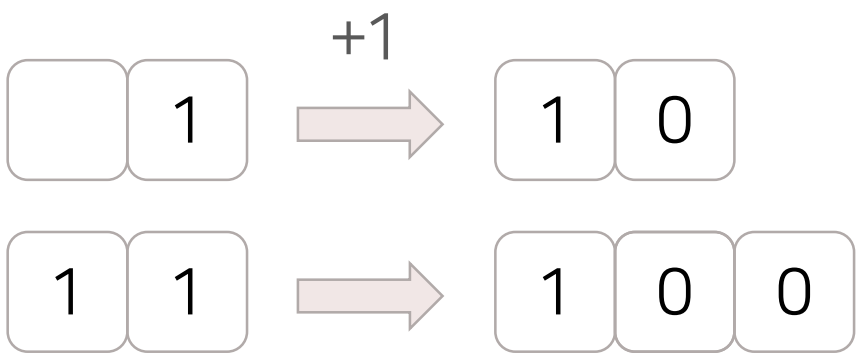
10진수

0 1 2 3 4 5 6 7 8 9
10가지 수로 숫자를 표현



2진수

0 1
2가지 수로 숫자를 표현



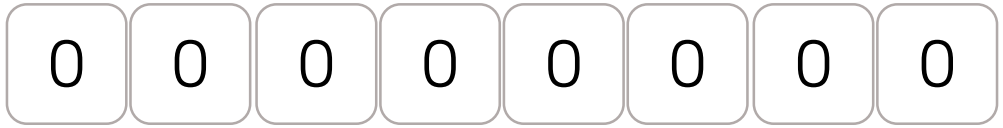
2 2진수의 이해 p-48

2진수	10진수
0	0
1	1
10	2
11	3
100	4
101	5

2진수의 이해 p-48

1 bit (비트)

8 bit (비트) -> 1 byte (바이트)



기타 단위 변환
1000 byte (바이트) = 1 kilo byte (키로 바이트)
1000 kilo byte (키로 바이트) = 1 mega byte (메가 바이트)
1000 mega byte (메가 바이트) = 1 giga byte (기가 바이트)

문자와 정수의 bit표현 p-50

자료형	형태	크기	값의 범위
char	정수(문자)	1 바이트	-128 ~ 127
int	정수	4 바이트	-2,147,483,648 ~ 2,147,483,647

char형 : 1 byte (바이트)

0	0	0	0	0	0	0	0

int형 : 4 byte (바이트)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

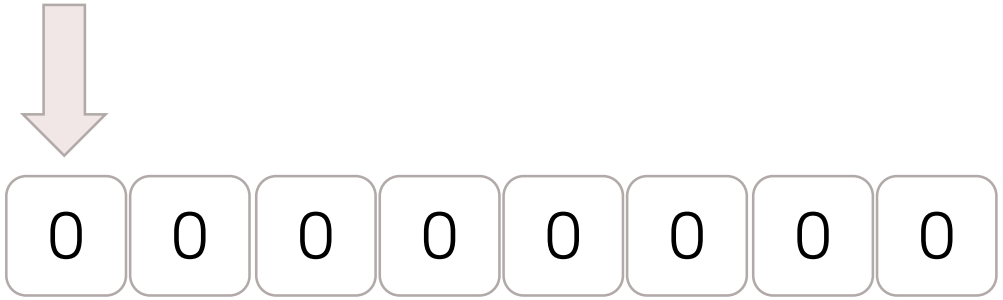
sizeof 함수 p-36

자료형	크기
char	1 바이트
short	2 바이트
int	4 바이트
long	4 바이트
long long	8 바이트
float	4 바이트
double	8 바이트
long double	8 바이트 이상

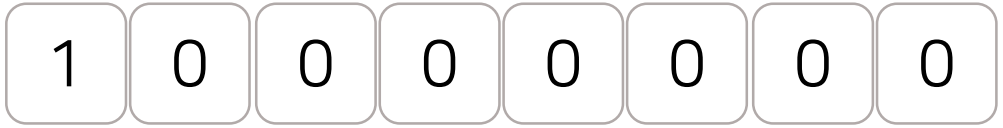
```c
#include <stdio.h>

int main(void) {
    char ch = 'A';
    int num = 10;

    printf("size of ch : %d₩n", sizeof(ch));
    printf("size of num : %d₩n", sizeof(num));
}
```

2 음의 정수의 bit표현 p-50

± : 부호비트



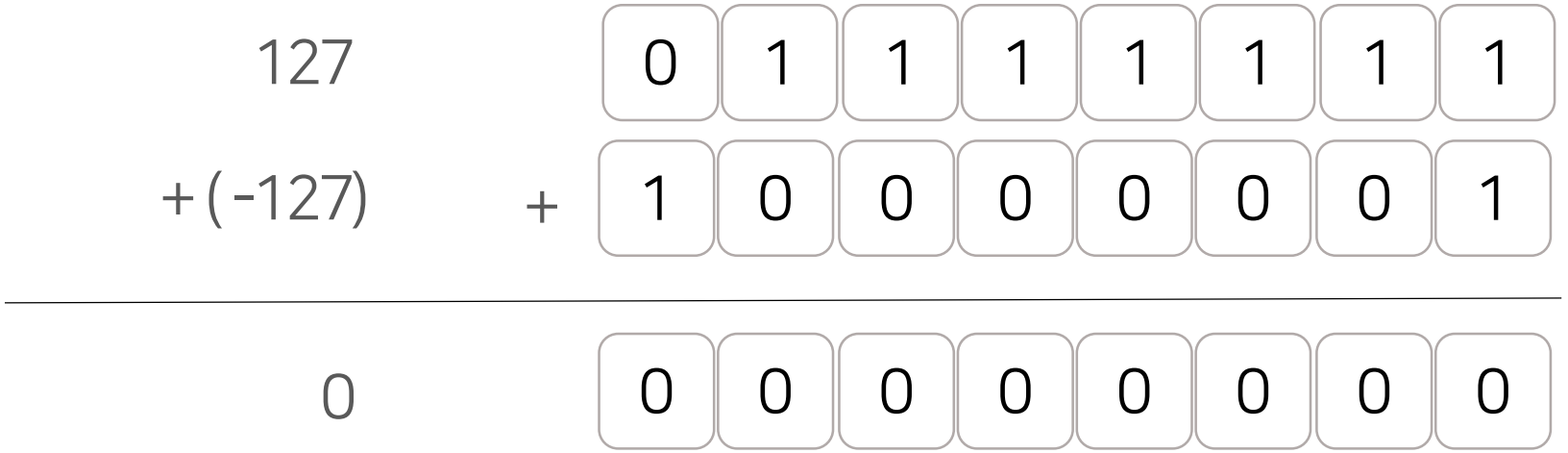
부호 비트가 1 이면 음수



2 음의 정수의 bit표현 p-50

어떤수의 음수는 더해서 0이되는 수

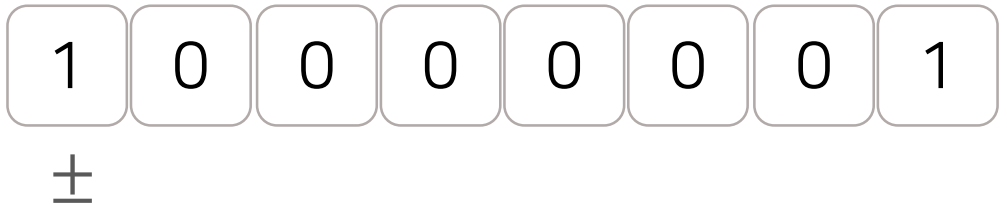
예) 127 – 127 = 0
127 + (-127) = 0



unsigned 형 p-54

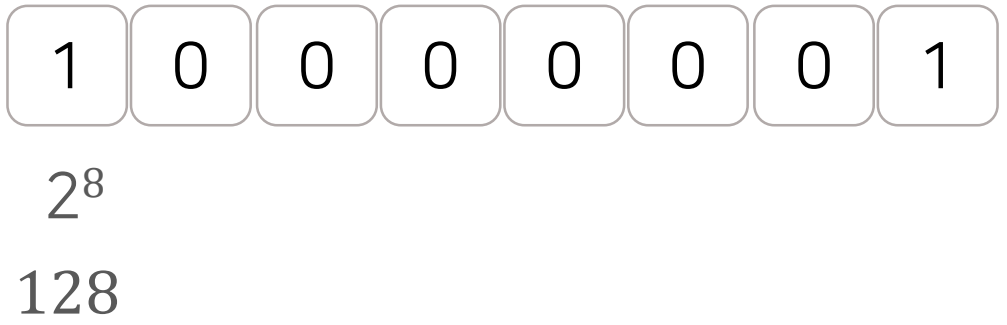
signed 형 (부호비트 o)

-127



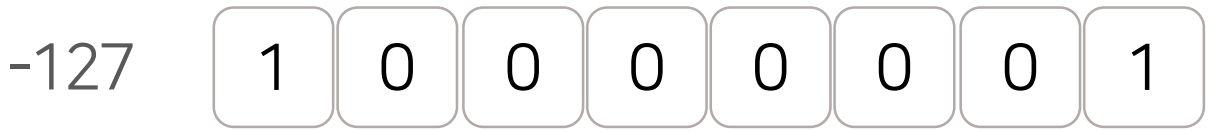
unsigned 형 (부호비트 x)

129



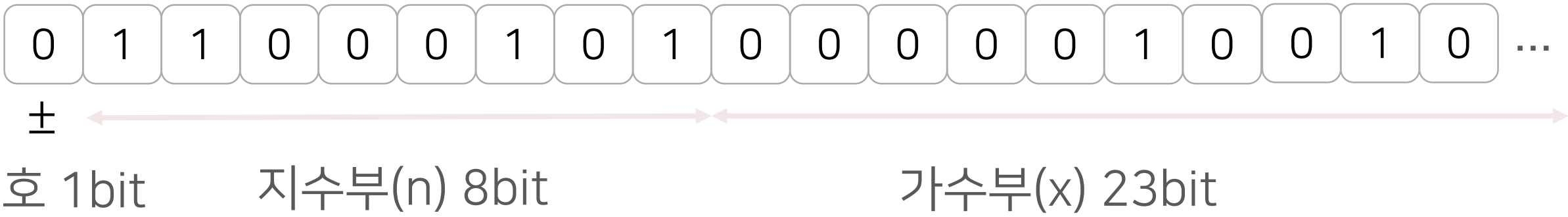
unsigned 형 p-54

```c
int main(void) {
    char signed_ch = -127;
    unsigned char unsigned_ch = -127;

    printf("signed : %d ₩n", signed_ch);
    printf("unsigned : %d ₩n", unsigned_ch);
}
```



2 실수의 bit표현 p-52

실수는 1.xxxxx 와 2^n 의 곱으로 표현 - 부동소수점 방식



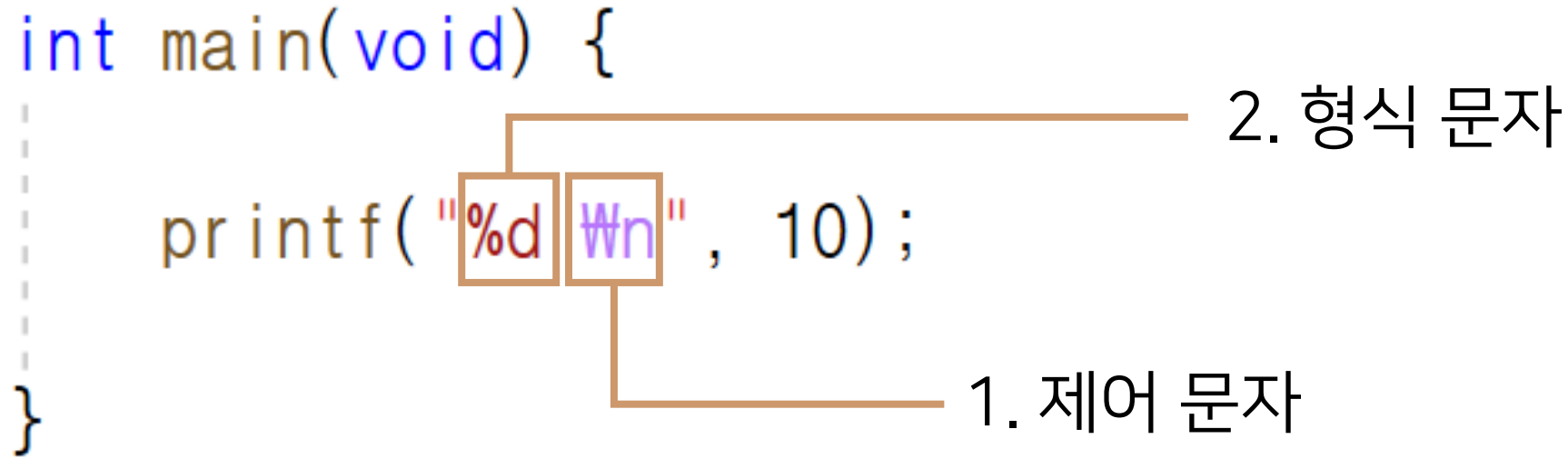
float (4byte, 32bit) 의 지수부와 가수부

3.

기본 입출력

3 printf 함수의 이해 p-60

printf 는 크게 두가지의 특수문자가 있습니다.



3 printf 함수의 제어문자 p-61

제어 문자	의미
₩n	개행 (New Line) – 새로운 줄 추가
₩t	수평 탭 – 일정간격을 수평으로 추가
₩v	수직 탭 – 일정간격을 수직으로 추가
₩b	백스페이스(Backspace) – 왼쪽으로 한칸이동
₩r	캐리지 리턴(Carriage Return) – 첫번째 칸으로 이동
₩a	경고음 (Alarm)
₩'	작은따옴표 출력
₩"	큰 따옴표 출력
₩?	물음표 출력
₩₩	백슬레시(₩) 기호 출력
₩f	폼 피드(Form Feed)

3 printf 함수의 제어문자 p-62

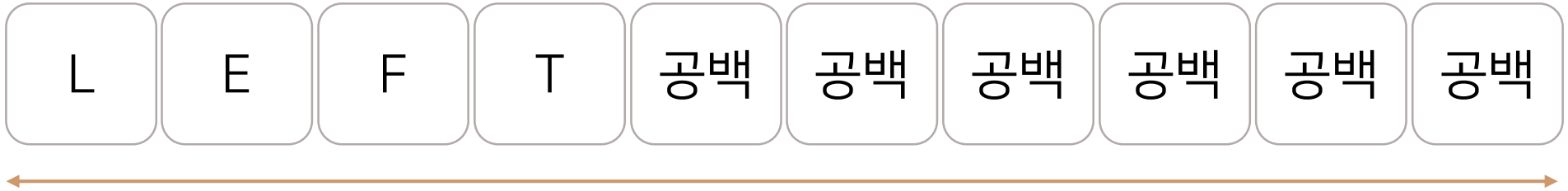
형식 문자	자료형	출력 형태
%d	char, short, int	부호 있는 10진수 정수
%u	unsigned int	부호 없는 10진수 정수
%o	unsigned int	부호 없는 8진수 정수
%x	unsigned int	부호 없는 16진수 정수
%f	float, double	10진수 방식의 부동소수점 실수
%e	float, double	지수 방식의 부동소수점 실수
%g	float, double	값에 따라 %f와 %e 중 선택
%c	char, short, int	값에 대응하는 문자
%s	char *, char []	문자열
%p	void *	포인터의 주솟값
%%		% 기호를 출력

3 형식문자로 출력 형태 정의하기 p-64

N	O	R	M	A	L

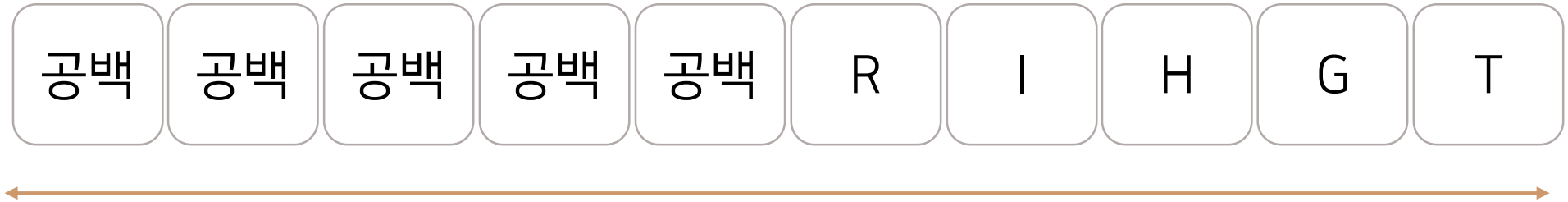
```c
int main(void) {
    printf("%s", "NORMAL");
    printf("%s", "END");
}
```

3 형식문자로 출력 형태 정의하기 p-64



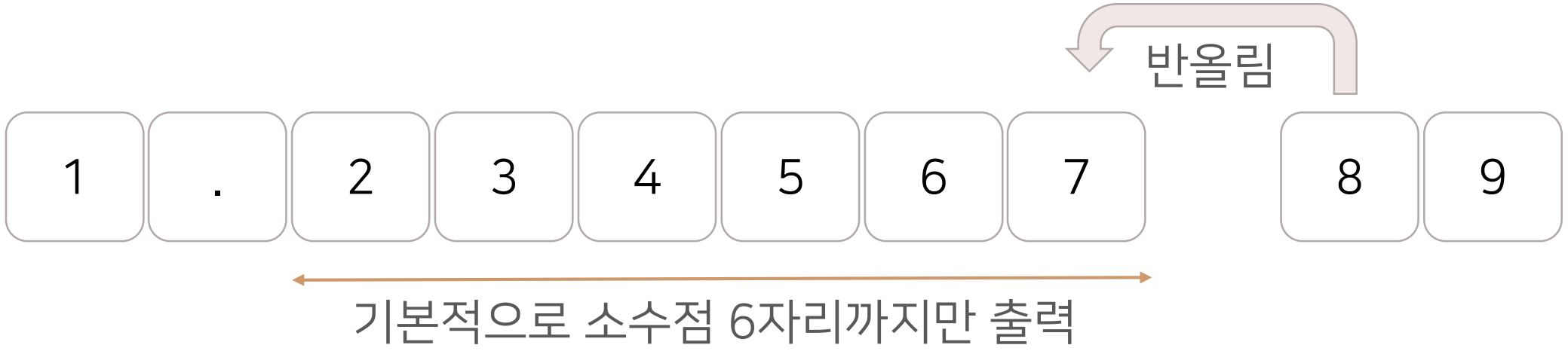

```c
int main(void) {
    printf("%-10s", "LEFT");
    printf("%s", "END");
}
```

3 형식문자로 출력 형태 정의하기 p-64



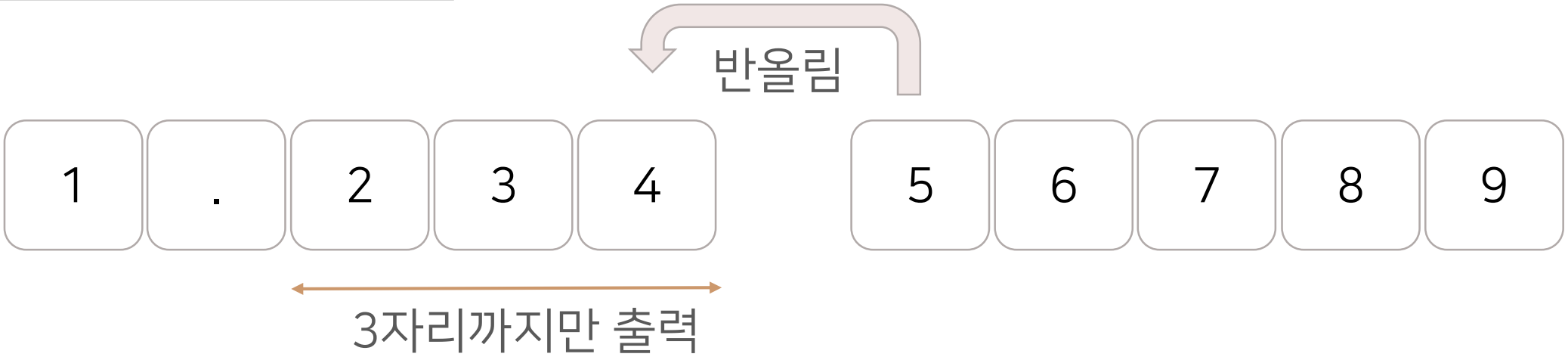

```c
int main(void) {
    printf("%10s", "RIGHT");
    printf("%s", "END");
}
```

3 형식문자로 출력 형태 정의하기 p-64



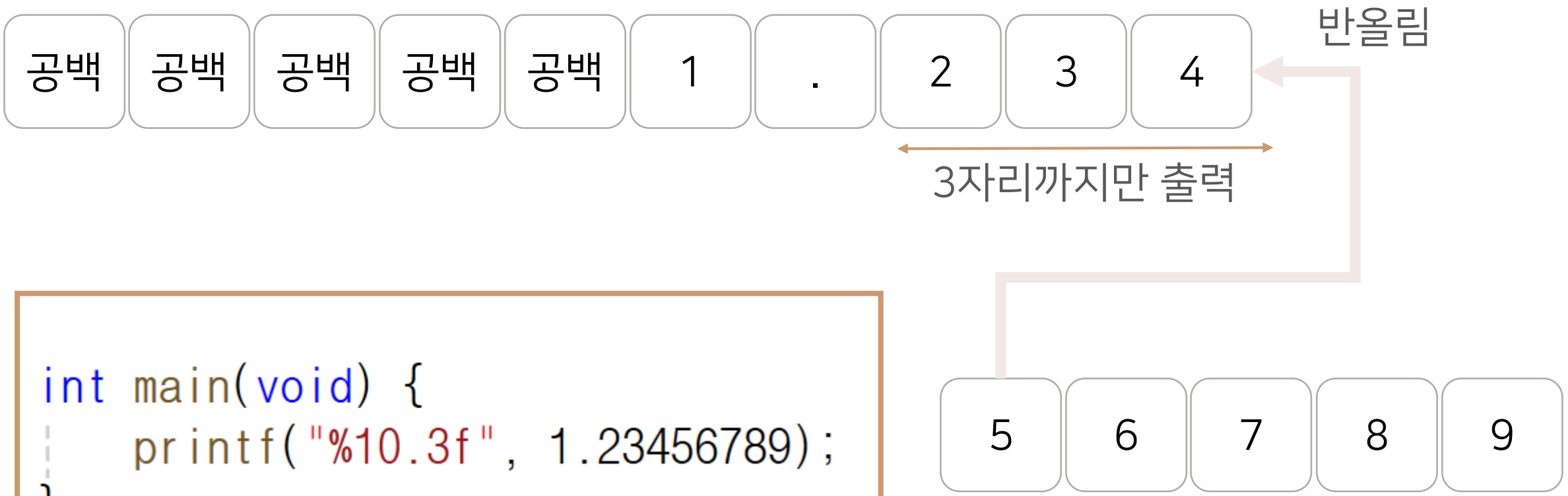

```c
int main(void) {
    printf("%f", 1.23456789);
}
```

3 형식문자로 출력 형태 정의하기 p-64




```c
int main(void) {
    printf("%.3f", 1.23456789);
}
```

3 형식문자로 출력 형태 정의하기 p-64




```c
int main(void) {
    printf("%10.3f", 1.23456789);
}
```

3 scanf_s의 이해 p-66

```
int main(void) {

    int num;

    scanf_s("%d", &num);

    return 0;
}
```

&변수명

형식 문자

3 scanf_s의 이해 p-66

```c
int main(void) {
    char ch;
    int inum;
    float fnum;

    printf("문자입력 : ");
    scanf_s("%c", &ch, 1);
    printf("정수입력 : ");
    scanf_s("%d", &inum);
    printf("실수입력 : ");
    scanf_s("%f", &fnum);

    printf("%c, %d, %f₩n", ch, inum, fnum);
}
```

3 printf 와의 형태 문자 차이

```c
int main(void) {
    double dnum;
    printf("실수를 입력하세요 : ");
    scanf_s("%lf", &dnum);
    printf("%f₩n", dnum);
}
```

double 형은 printf 에서는 %f

scanf_s 에서는 %lf 입니다.

3 다양한 입력 형태 p-68

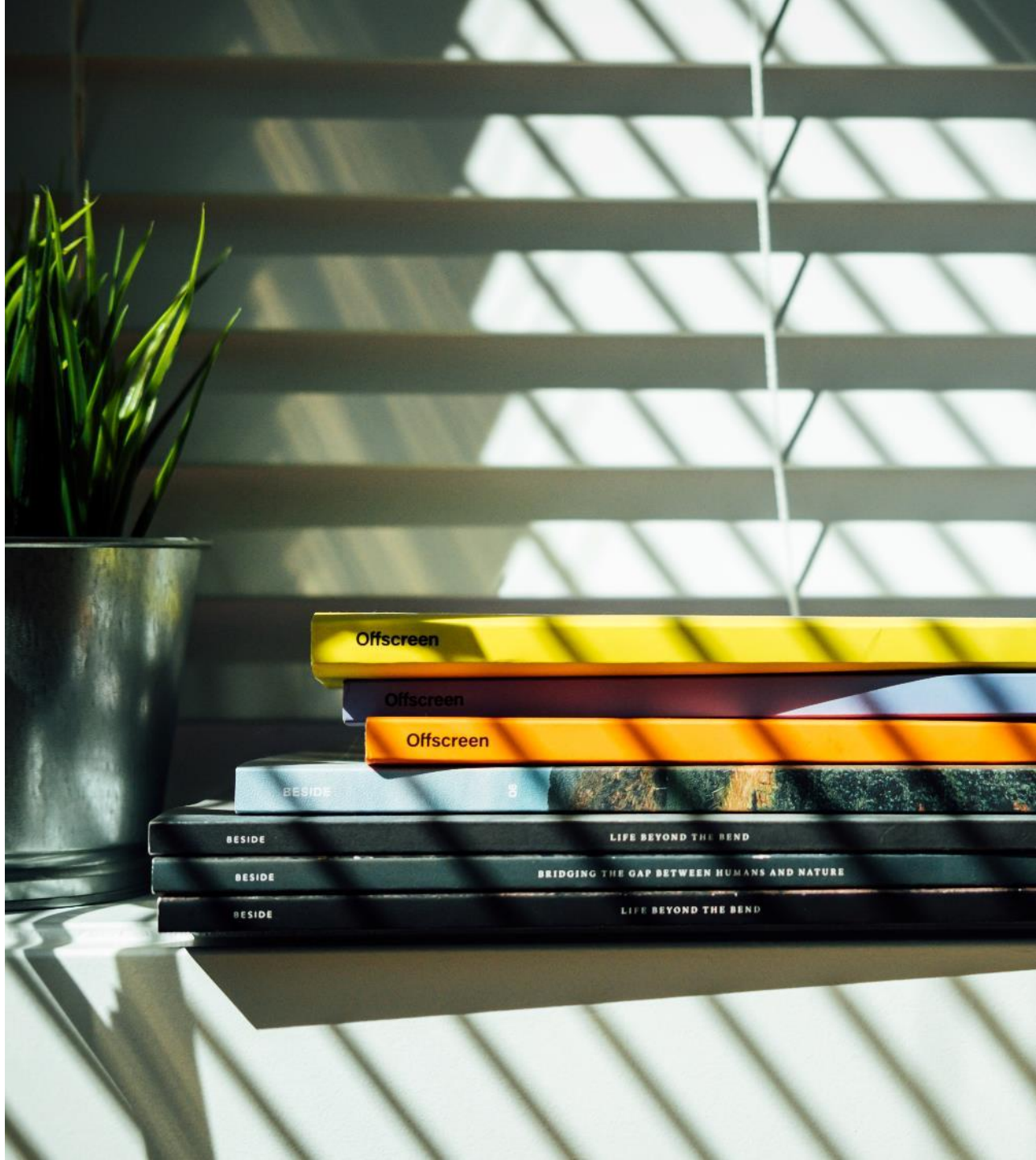
```c
int main(void) {
    int num1, num2, num3;

    printf("세개의 정수 입력: ");
    scanf_s("%d %d %d", &num1, &num2, &num3);
    printf("입력된 정수들 : %d %d %d₩n", num1, num2, num3);

}
```

3 scanf 의 엔터 입력 문제

```c
int main(void) {
    char ch1, ch2;
    printf("첫번째 문자를 입력하세요 : ");
    scanf_s("%c", &ch1, 1);
    printf("첫번째 입력된 문자 : %c₩n", ch1);
    rewind(stdin); // 입력버퍼를 초기화

    printf("두번째 문자를 입력하세요 : ");
    scanf_s("%c", &ch2, 1);
    printf("두번째 입력된 문자 : %d₩n", ch2);
}
```

4.

연산자


Offscreen
Offscreen
Offscreen
BESIDE
BESIDE
LIFE BEYOND THE BEND
BESIDE
BRIDGING THE GAP BETWEEN HUMANS AND NATURE
BESIDE
LIFE BEYOND THE BEND

4 산술 연산자 p-74

연산자	기능	사용 예
+	두 피연산자 값을 더합니다.	5 + 3
-	왼쪽 값 에서 오른쪽 값을 뺍니다.	10 - 4
*	두 값을 곱합니다.	6 * 8
/	왼쪽 값을 오른쪽 값으로 나눕니다.	9 / 3
%	왼쪽 값을 오른쪽 값으로 나누었을 때의 나머지를 구합니다.	9 % 2 (피연산자가 정수만 가능)

4 산술 연산자 p-74

```c
int main(void) {
    int num1 = 7, num2 = 3;
    printf("%d + %d = %d₩n", num1, num2, num1 + num2);
    printf("%d - %d = %d₩n", num1, num2, num1 - num2);
    printf("%d * %d = %d₩n", num1, num2, num1 * num2);
    printf("%d / %d = %d₩n", num1, num2, num1 / num2);
    printf("%d %% %d = %d₩n", num1, num2, num1 % num2);
    return 0;
}
```

4 산술 연산자 p-74

```c
int main(void) {
    int num1, num2;
    printf("두 정수를 입력하세요: ");
    scanf_s("%d %d", &num1, &num2);
    printf("%d + %d = %d₩n", num1, num2, num1 + num2);
    printf("%d - %d = %d₩n", num1, num2, num1 - num2);
    printf("%d * %d = %d₩n", num1, num2, num1 * num2);
    printf("%d / %d = %d₩n", num1, num2, num1 / num2);
    printf("%d %% %d = %d₩n", num1, num2, num1 % num2);
    return 0;
}
```

4 0으로 나누기

```c
int main(void) {
    printf("%d", 10 / 0);
    return 0;
}
```

0 으로 나누기는 대부분의 프로그램에서 문제를 일으키기 때문에 주의 !!

4 나눈 결과값이 실수가 되려면

```c
int main(void) {
    int num1 = 7;
    float num2 = 3;
    float result = num1 / num2;
    printf("%f₩n", result);
    printf("%f", (float)7 / 3);
    // 두 피연산자 중 하나가 float 형태면
    // 결과가 실수가 됩니다.
}
```

4 도형 넓이 구하기

```c
// 삼각형 사각형 넓이 구하기
int main(void) {
    float x, y;
    printf("가로와 세로값을 입력하세요 : ");
    scanf_s("%f %f", &x, &y);
    printf("가로 : %.3f, 세로 : %.3f₩n", x, y);
    printf("사각형의 넓이 : %.3f₩n", x * y);
    printf("삼각형의 넓이 : %.3f₩n", x * y * 0.5);

}
```

4 도형 넓이 구하기

```c
// 원의둘레, 넓이 구하기
int main(void) {
    const float pi = 3.14;
    float r;
    printf("반지름의 길이를 입력하세요 : ");
    scanf_s("%f", &r);
    printf("반지름 길이 : %g₩n", r);
    printf("원의 둘레 : %g₩n", 2 * pi * r);
    printf("원의 넓이 : %g₩n", pi * r * r);
}
```

4 단위 변환하기

```c
// 달러를 원으로 표시하기
int main(void) {
    float dollor_one;
    float dollor;

    printf("1달러의 환율을 입력하세요 : ");
    scanf_s("%f", &dollor_one);
    printf("환산하고자 하는 달러를 입력하세요 : ");
    scanf_s("%f", &dollor);
    printf("%.2f 달러 = %.2f 원₩n(달러-원 환율 : %.2f원)",
        dollor, dollor * dollor_one, dollor_one);
}
```