Fundamentals of Computer System
- Data and C (2) 25 March 2009

민기복

Ki-Bok Min, PhD

서울대학교 에너지자원공학과 조교수

Assistant Professor, Energy Resources Engineering

SEOUL NATIONAL UNIVERSITY

Lecture 3 -Data and C(1)
Summary of last lecture

SEOUL NATIONAL UNIVERSITY

- Multiple functions
- Interactive program using scanf()
- C data types – integer, char, float
- Overflow

5. Taking another step
Multiple functions

SEOUL NATIONAL UNIVERSITY

prototype

Function call

butler() executed here!!!

Function definition

```
/* two_func.c -- 하나의 파일에서 두 개의 함수를 사용하는 프로그램 */
#include <stdio.h>
void butler(void);          /* ISO/ANSI C 함수 프로토타입 */
int main(void)
{
    printf("butler 함수 밖에 있느냐?\n");
    butler();
    printf("그래. 차 한 잔 내오고, CD-ROM 드라이브도 가져오너라.\n");

    return 0;
}
void butler(void)           /* 함수 정의의 시작 */
{
    printf("부르셨습니까? 주인님!\n");
}
```

- butler() function appeared three times: prototype, function call, function definition
- void butler(void);

no return value

no argument

- Prototype: - a declaration telling the compiler that you are using a function.
  - specifies properties of the function
- The location of function definition does not matter. But it's nice to put main() first.

1. A sample program (gold.c)
an interactive program using scanf()

SEOUL NATIONAL UNIVERSITY

```c
/* gold.c  -- your weight in gold */
#include <stdio.h>
int main(void)
{
    float weight;     /* user weight            */
    float value;      /* gold equivalent      */

    printf("Are you worth your weight in gold?\n");
    printf("Let's check it out.\n");
    printf("Please enter your weight in kg: ");

/* get input from the user                     */
    scanf("%f", &weight);
/* assume gold is $1000.0 per ounce           */
/* 35.2423 converts kg to ounces */
    value = 1000.0 * weight * 35.2423;
    printf("Your weight in gold is worth $%.2f.\n", value);
    printf("You are easily worth that! If gold prices drop,\n");
    printf("eat more to maintain your value.\n");

    return 0;
}
```

- Keyboard input
- %f for floating-point number
- &weight to assign input value to weight
- Why &? Has to do with address.

- This program is interactive (대화식) – computer asks you for information and uses the number you enter.
- Compare this with “hello world” program – a significant improvement.

Data Types
C data keywords

the way the data is stored

integer type (정수형)

floating-point type (실수형)

Original K&R Keywords	C90 Keywords	C99 Keywords
int	signed	_Bool
long	void	_Complex
short		_Imaginary
unsigned		
char		
float		
double		

Type sizes

SEOUL NATIONAL UNIVERSITY

Type	Macintosh	Linux on a PC	IBM PC WINDOWS	ANSI C (minimum)
char	8	8	8	8
int	32	32	32	16
short	16	16	16	16
long	32	32	32	32
long long	64	64	64	64

Type	Macintosh	Linux on a PC	IBM PC WINDOWS	ANSI C (minimum)
float	6 -37 ~ 38	6 -37 ~ 38	6 -37 ~ 38	6 -37 ~ 37
double	18 -4931 ~ 4932	15 -307 ~ 308	15 -307 ~ 308	10 -37 ~ 37
long double	18 -4931 ~ 4932	18 -4931 ~ 4932	18 -4931 ~ 4932	10 -37 ~ 37

Number of significant digit (유효숫자 자리수)

Exponent range (10진수 지수범위)

Basic Data Type

Max/Min range (int for IBM PC/WINDOWS – 32bit)

SEOUL NATIONAL UNIVERSITY

0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	…	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
+/-	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	…	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}

- $a(r^n-1)/(r-1) = 1(2^{31}-1)/(2-1)=2^{31}-1$ =2,147,483,647→ maximum
- -2,147,483,648→ minimum
- Minimum = - maximum -1 (because negative integer is stored as two's complement (2의 보수)). Note that there is only one expression for zero.
- 2의 보수 = 1의 보수 + 1
- Ex) 2 bit system

0/1	0/1
+/-	2^{0}

Basic Data Type
Overflow – an example

SEOUL NATIONAL UNIVERSITY

```c
/* toobig.c-exceeds maximum int size on our system */
#include <stdio.h>
int main(void)
{
    int i = 2147483647;
    unsigned int j = 4294967295;

    printf("%d %d %d\n", i, i+1, i+2);
    printf("%u %u %u\n", j, j+1, j+2);

    return 0;
}
```

```
C:\Windows\system32\cmd.exe
2147483647 -2147483648 -2147483647
4294967295 0 1
계속하려면 아무 키나 누르십시오 . . .
```

Today

SEOUL NATIONAL UNIVERSITY

- char type
- float type
- Miscellaneous
  - Matching Arguments
  - Escape sequences
  - Matching data types

Basic Data Type
Character type

SEOUL NATIONAL UNIVERSITY

- Char type: characters (letters & punctuation)
  - Computer uses a numerical code (certain integer represent certain characters). E.g., ASCII (American Standard Code for Information Interchange) code
  - But technically it is integer
  - Because it actually stores integers
  - 65 represents upper case A. To store A, you store 65.

Decimal	Octal	Hex	Binary	Character	ASCII Name
45	055	0x2d	00101101	-	
46	056	0x2e	00101110	.	
47	057	0x2f	00101111	/	
48	060	0x30	00110000	0	
49	061	0x31	00110001	1	
50	062	0x32	00110010	2	
51	063	0x33	00110011	3	
52	064	0x34	00110100	4	
53	065	0x35	00110101	5	
54	066	0x36	00110110	6	
55	067	0x37	00110111	7	
56	070	0x38	00111000	8	
57	071	0x39	00111001	9	
58	072	0x3a	00111010	:	
59	073	0x3b	00111011	;	
60	074	0x3c	00111100	<	
61	075	0x3d	00111101	=	
62	076	0x3e	00111110	>	
63	077	0x3f	00111111	?	
64	0100	0x40	01000000	@	
65	0101	0x41	01000001	A	
66	0102	0x42	01000010	B	
67	0103	0x43	01000011	C	
68	0104	0x44	01000100	D	
69	0105	0x45	01000101	E	
70	0106	0x46	01000110	F	

Basic Data Type
Character type (ASCII code)

SEOUL NATIONAL UNIVERSITY

- Standard ASCII code: 0 ~ 127 < 8 bit
- Commercial Unicode Initiative: 16 bit or 32 bit

Basic Data Type (Char)
Declaration

SEOUL NATIONAL UNIVERSITY

- Create three variables: response, itable, latan

```
char response;
char itable, latan;
```

- Do we need to memorize ASCII code to assign character? No!! → use ‘ ’

```
char grade = 'A'
```

Character constant (문자상수)

```
grade = 'A';        /* OK*/
grade = A;          /* No! compiler think of A as a name of variables */
grade = "A";        /* No! compiler think of A as a string */
grade = 65;         /* OK for ascii but poor style */
```

Will be covered later

Basic Data Type (Char)
Nonprinting characters

SEOUL NATIONAL UNIVERSITY

- Nonprinting characters: backspace, enter, speaker beep…

  1) use ASCII code,

     ex) char beep = 7 ; // beeping

  2) use escape sequence

     ex) char nerf = '\n';

Basic Data Type (Char)
Nonprinting characters

SEOUL NATIONAL UNIVERSITY

- Nonprinting characters: backspace, enter, speaker beep

  1) Use ASCII code

     ex) char beep = 10; // new line

  2) Use escape sequence

     ex) char nerf = '\n';

If you want to print this

He said, "a \ is a backslash."

printf("He said, \"a \\ is a backslash.\"\n");

Use this

sequence	Meaning
\b	Back space
\n	New line
\t	Horizontal tab
\\	Backslash (\)
\'	Single quote
\"	Double quote
\?	Question mark

Basic Data Type (Char)
Number (수) & Number character (숫자상수)

SEOUL NATIONAL UNIVERSITY

- Number vs. Number character
  - Ex) character 4 is represented by ascii 52 or ‘4’

Basic Data Type (Char)
Printing Characters

SEOUL NATIONAL UNIVERSITY

- Use %c
- As char is stored as 1 byte integer, if you print the value of a char variable with usual %d, you get an integer

```c
/* charcode.c-displays code number for a character */
#include <stdio.h>
int main(void)
{
    char ch;

    printf("Please enter a character.\n");
    scanf("%c", &ch);   /* user inputs character */
    printf("The code for %c is %d.\n", ch, ch);

    return 0;
}
```

scanf() function fetches the character you typed, & causes the character to be assigned to the variable ch

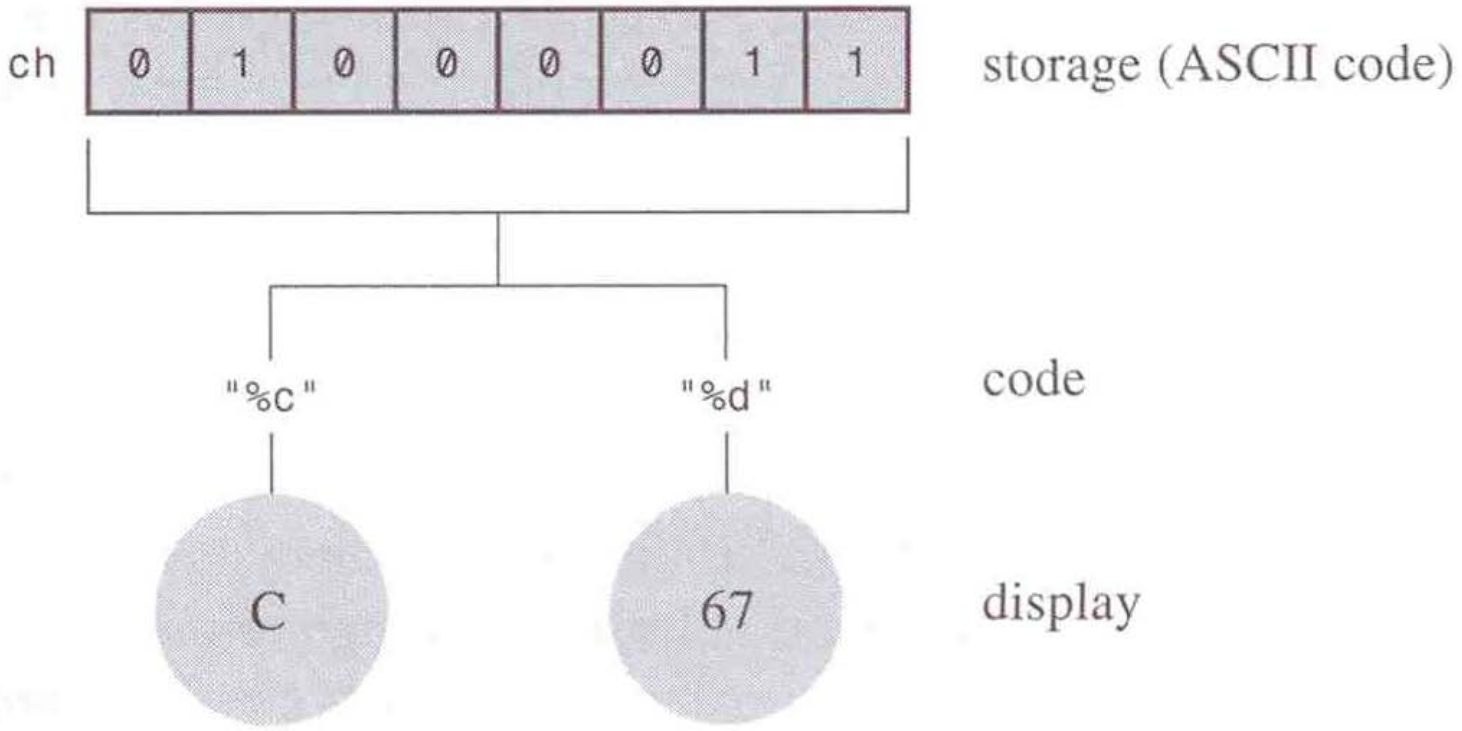
```
C:\Windows\system32\cmd.exe
Please enter a character.
A
The code for A is 65.
계속하려면 아무 키나 누르십시오 . . .
```

printf() has to do with display (not how it is stored)

Basic Data Type (Char)
Printing Characters

SEOUL NATIONAL UNIVERSITY

- Data Display versus Data Printing



Basic Data Type (_Bool type)

SEOUL NATIONAL UNIVERSITY

- Requires 1 bit of memory: 1 (true) or 0 (false)

Basic Data Type (floating-point number, 실수형)

SEOUL NATIONAL UNIVERSITY

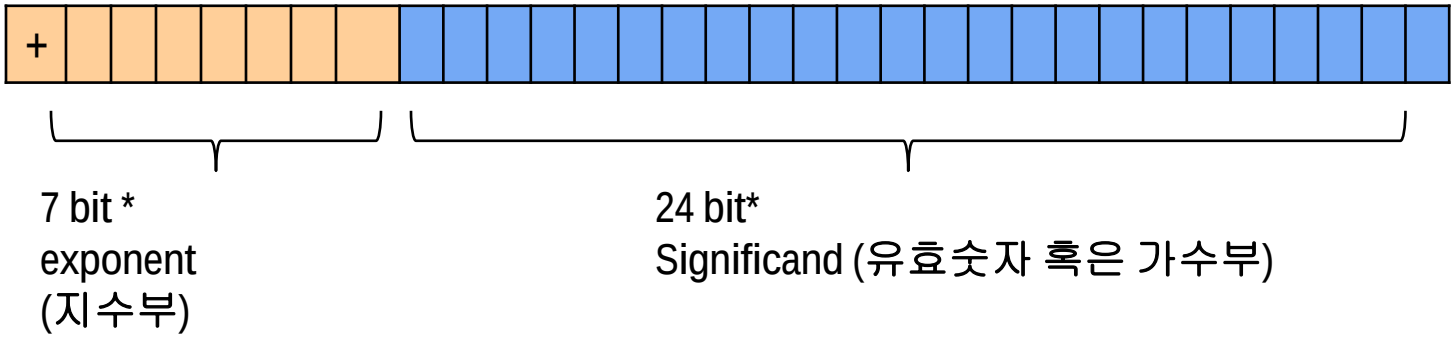
- integer is not enough, e.g., mathematical computing
- Floating point number: **float, double, long double**
- Similar to **real** in FORTRAN

Number	Scientific Notation (과학적 표기)	Exponential notation (지수표기)
1,000,000,000	1.0×10^{9}	1.0e9
123,000	1.23×10^{5}	1.23e5
322.56	3.2256×10^{2}	3.2256e2
0.000056	5.6×10^{-5}	5.6e-5

Basic Data Type (floating-point number)

SEOUL NATIONAL UNIVERSITY

- Floating point: 32 bit $\pm 3.4\times10^{-38} \sim \pm 3.4\times10^{38}$



- Double: 64 bit $\pm 1.7\times10^{-308} \sim \pm 1.7\times10^{308}$
  - Use additional 32 bit for significand or exponent
- Declaration of variables is similar to integer

```
float noah, jonah;          // declare noah & jonah
float planck = 6.63e-34;    // initialize planck
```

*can vary slightly depending on machines

Floating-point number
floating point constants

SEOUL NATIONAL UNIVERSITY

- Valid floating-point constants

  3.14159e+12

  3.14159

  3.14159E12

  3e12

  .14159e-2

  100.

- Not valid

  1.56 E+12

  there is a space

Floating-point number
floating point constants

SEOUL NATIONAL UNIVERSITY

- By default, the compiler assumes floating-points constants are **double** precision.
  - float some; some = 4.0 * 2.0;
    - 4.0 & 2.0 stored as double (using typically 64 bit),
    - the product is calculated using double precision
    - Answer trimmed to regular **float** size
    - This ensures a greater precision but slows down a program
- Use f or F suffice to make the compiler treat a floating-point constant as type **float**
  - float some; some = 4.0f * 2.0F;

Floating-point number
floating point constants

SEOUL NATIONAL UNIVERSITY

- Or use l or L suffix to treat a floating-point constant as type long double
  - float some; some = 4.0L * 2.0L;

Floating-point number
Printing (format specifier)

SEOUL NATIONAL UNIVERSITY

- %f :print type **float** and **double** using decimal notation
- %e : - using exponential notation

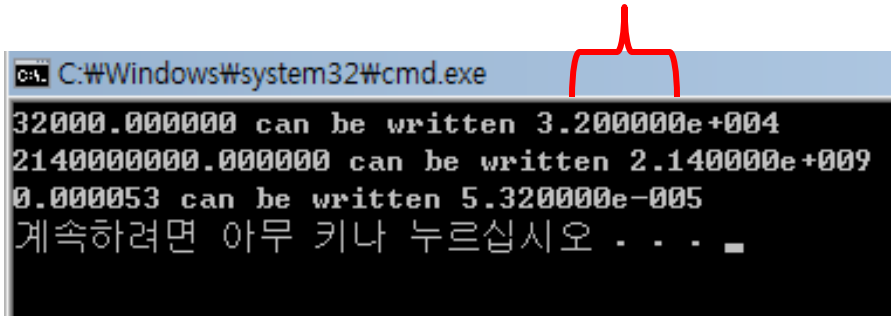
전역 범위)

```c
/* showf_pt.c -- displays float value in two ways */
#include <stdio.h>
int main(void)
{
    float aboat = 32000.0;
    double abet = 2.14e9;
    long double dip = 5.32e-5;

    printf("%f can be written %e\n", aboat, aboat);
    printf("%f can be written %e\n", abet, abet);
    printf("%f can be written %e\n", dip, dip);

    return 0;
}
```

Number of places to the right of the decimal
(소수점 아래 자리수)



Floating-point number
Overflow & Underflow

SEOUL NATIONAL UNIVERSITY

- Overflow: when a number is too big INF
- Underflow: e.g., 0.1234E-10 /10 → 0.0123E-10,
  - we lost a digit,


```c
/* showf_pt.c -- displays float value in two ways */
#include <stdio.h>
int main(void)
{
    float toobig = 3.4e38 * 100.0f;
    printf("%e\n", toobig);

    return 0;
}
```

C:₩Windows₩system32₩cmd.exe

```
1.#INF00e+000
계속하려면 아무 키나 누르십시오 . . .
```

```
ㅌ: ch3, 구성: Debug Win32 ------

:) computer programming₩c 기초플러스 자료₩english version₩ch03₩showf_pt.c(5) : warning C4056: 부동 소수점 상수 산술 연산에서 오버플로가 발생했습니다.
:) computer programming₩c 기초플러스 자료₩english version₩ch03₩showf_pt.c(5) : warning C4756: 상수 산술 연산에서 오버플로가 발생했습니다.
:) computer programming₩c 기초플러스 자료₩english version₩ch03₩showf_pt.c(5) : warning C4756: 상수 산술 연산에서 오버플로가 발생했습니다.
```

Type sizes

SEOUL NATIONAL UNIVERSITY

Type	Macintosh	Linux on a PC	IBM PC WINDOWS	ANSI C (minimum)
char	8	8	8	8
int	32	32	32	16
short	16	16	16	16
long	32	32	32	32
long long	64	64	64	64

Type	Macintosh	Linux on a PC	IBM PC WINDOWS	ANSI C (minimum)
float	6 -37 ~ 38	6 -37 ~ 38	6 -37 ~ 38	6 -37 ~ 37
double	18 -4931 ~ 4932	15 -307 ~ 308	15 -307 ~ 308	10 -37 ~ 37
long double	18 -4931 ~ 4932	18 -4931 ~ 4932	18 -4931 ~ 4932	10 -37 ~ 37

Number of significant digit (유효숫자 자리수)

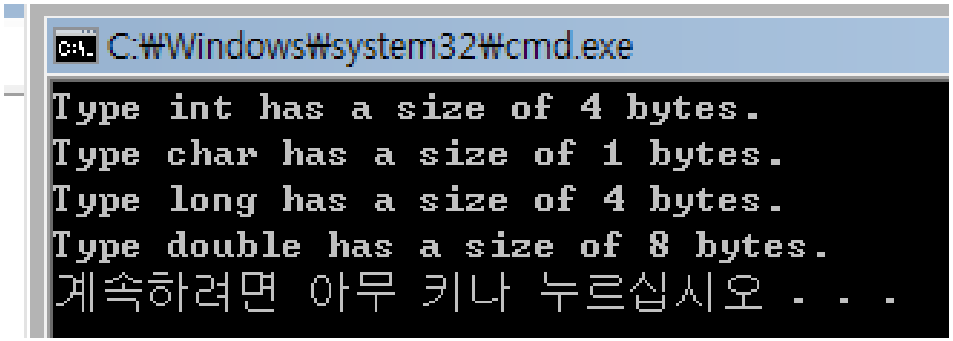
Exponent range (10진수 지수범위)

Type sizes

SEOUL NATIONAL UNIVERSITY

```c
/* typesize.c -- prints out type sizes */
#include <stdio.h>
int main(void)
{
/* c99 provides a %zd specifier for sizes */
    printf("Type int has a size of %u bytes.\n", sizeof(int));
    printf("Type char has a size of %u bytes.\n", sizeof(char));
    printf("Type long has a size of %u bytes.\n", sizeof(long));
    printf("Type double has a size of %u bytes.\n",
           sizeof(double));
    return 0;
}
```

Printf() statements is spread over two lines



Matching the constant types

SEOUL NATIONAL UNIVERSITY

- int apples = 3; /* good */
- int apples = 3.0; /* bad */
- int cost = 12.99; /* bad */ 12만 입력
- float pi = 3.1415926536; /* bad */ 3.141593만 출력

- Ex) c = ???

```
#include <stdio.h>
int main(void)
{
int a = 2.99;
int b = 1.99;
int c = a + b;

    printf("result is %d ₩n", c);

    return 0;
}
```

Arguments

SEOUL NATIONAL UNIVERSITY

- Number of format specifier = number of arguments
- Specifier type = value type

```c
/* badcount.c -- incorrect argument counts */
#include <stdio.h>
int main(void)
{
    int f = 4;
    int g = 5;
    float h = 5.0f;

    printf("%d₩n", f, g);     /* too many arguments   */
    printf("%d %d₩n",f);      /* too few arguments    */
    printf("%d %f₩n", h, g); /* wrong kind of values */

    return 0;
}
```

```
4
4 0
0 0.000000
계속하려면 아무 키나 누르십시오 . . .
```

Result can be very wrong! 0 in this case.

- Simply using %d to display **float** value, or
- Simply using %f to display **integer** value print garbage value.
- In general, compiler doesn't show error message for this. – be careful.

Escape sequences

SEOUL NATIONAL UNIVERSITY

```c
/* escape.c -- uses escape characters */
#include <stdio.h>
int main(void)
{
    float salary;

    printf("\aEnter your desired monthly salary:");/* 1 */
    printf(" $_______\b\b\b\b\b\b\b");             /* 2 */
    scanf("%f", &salary);
    printf("\n\t$%.2f a month is $%.2f a year.", salary,
               salary * 12.0);                     /* 3 */
    printf("\rGee!\n");                            /* 4 */

    return 0;
}
```

\b : same as ‘backspace’ in keyboard

\t : same as tab (usually 9th column)

\r : position the cursor at the beginning of the current line

Summary

SEOUL NATIONAL UNIVERSITY

- **char** type ; use ASCII code,
- **float** type ; **float** (32bit), **double** (64bit),
- Miscellaneous
  - Matching Arguments
  - Escape sequences ; /b, /t, /r
  - Matching data types ; be careful