Fundamentals of Computer System
- Character strings & Formatted I/O

민기복

Ki-Bok Min, PhD

서울대학교 에너지자원공학과 조교수

Assistant Professor, Energy Resources Engineering

SEOUL NATIONAL UNIVERSITY

Summary of last lecture

SEOUL NATIONAL UNIVERSITY

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

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)

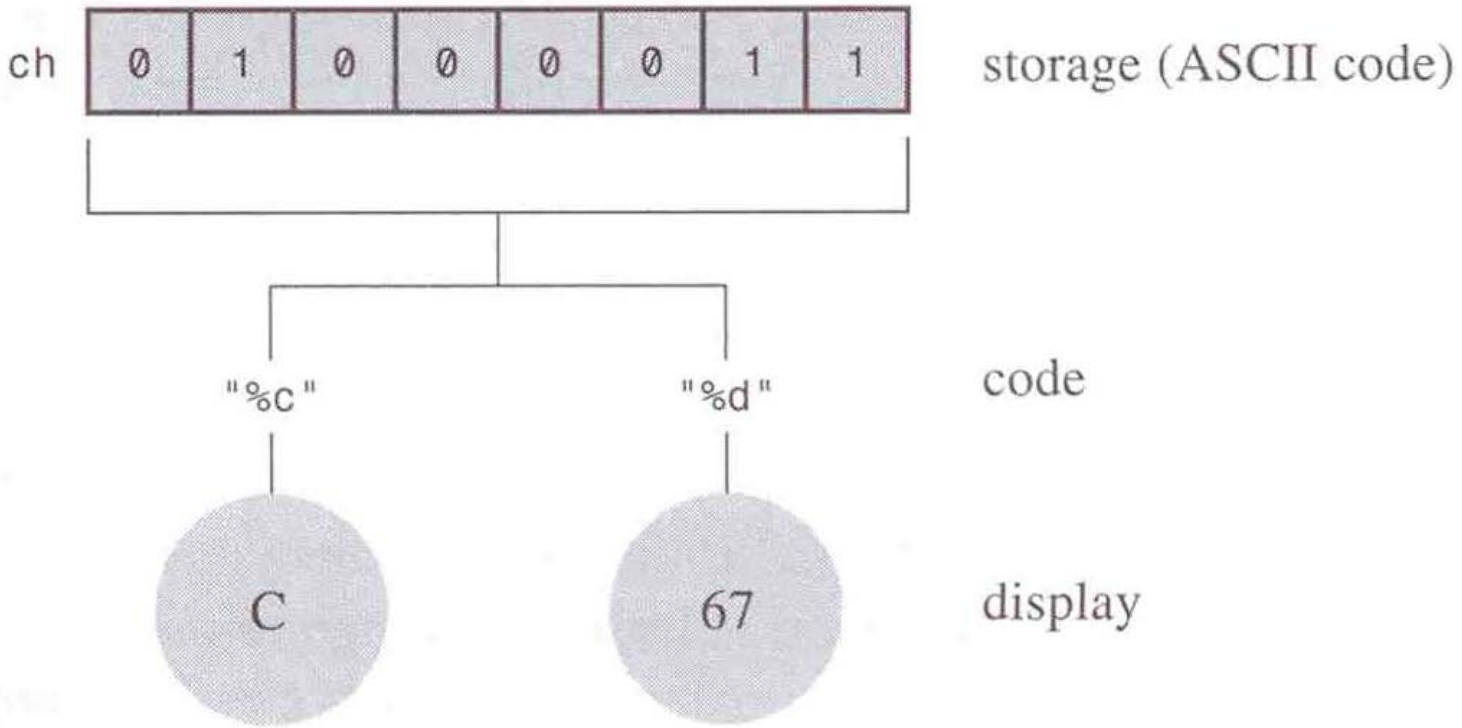
```
grade = A;        /* No! compiler think of A as a name of variables */
grade = "A";      /* No! compiler think of A as a string */
grade = 65;       /* OK for ascii but poor style */
```

Will be covered today!

Basic Data Type (Char)
Printing Characters

SEOUL NATIONAL UNIVERSITY

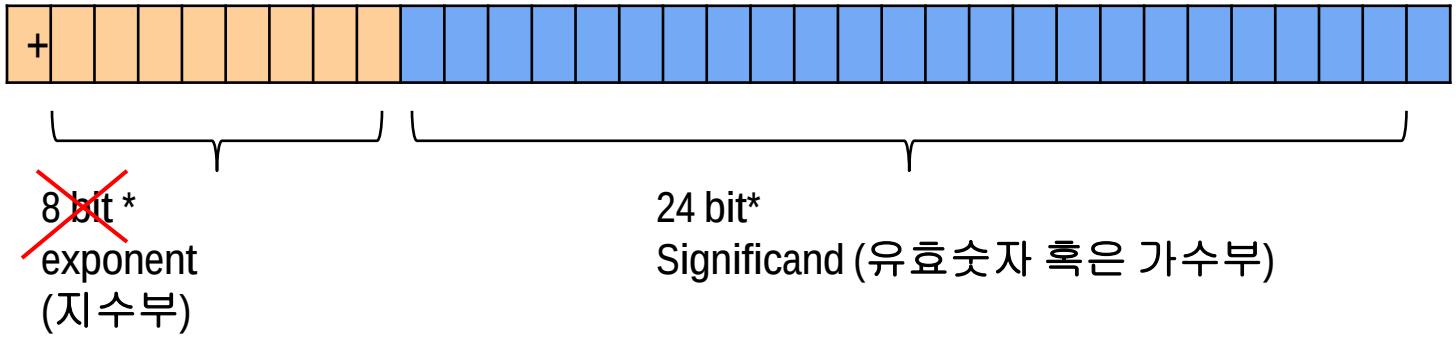
- Data Display versus Data Printing



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

Today

SEOUL NATIONAL UNIVERSITY

- character string
- printf()
  - Format specifier(포맷지정자), Format modifier(포맷변경자)
- scanf()
  - Format specifier, Format modifier
- Miscellaneous
  - Other header files
  - Manifest constant (명단함수)

Character strings (문자열)

SEOUL NATIONAL UNIVERSITY

```c
// talkback.c -- nosy, informative program
#include <stdio.h>
#include <string.h>       // for strlen( ) prototype
#define DENSITY 62.4      // human density in lbs per cu ft
int main( )
{
    float weight, volume,weight_p;
    int size, letters;
    char name[40];          // name is an array of 40 chars

    printf("What is the first name of your favorite player in WBC baseball teams?₩n");
    scanf("%s", name);
    printf("%s, How much does he weigh in kilogram?₩n", name);
    scanf("%f", &weight);
    size = sizeof name;
    letters = strlen(name);
    weight_p = weight / 0.454;
    volume = weight / DENSITY;
    printf("%s weighs %2.2f pound.₩n",
           name, weight_p);
    printf("His name has %d letters.₩n",
           letters);
    printf("Also and we have %d bytes to store it in.₩n", size);

    return 0;
}
```

명단상수(manifest constant)

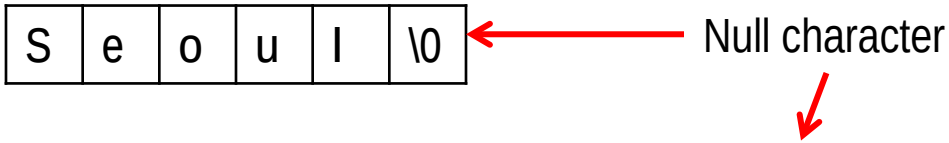
- Use array for character string
- Use %s
- Symbolic constant DENSITY
- Length of string: strlen()

character string (문자열)

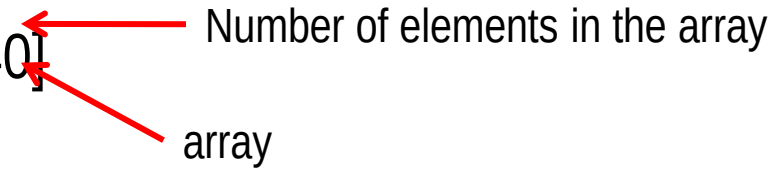
SEOUL NATIONAL UNIVERSITY

문자열: 널문자로 끝나는 하나의 단위로 취급되는 연속적인 문자

- character string: a series of one or more characters, an array
  - No special variable type – array of type **char**, use " "
  - Stored in adjacent memory cells – one character (1 byte) per cell,
  - 한글 한 글자는?
  - Null character (널 문자) to mark the end of a string



  - Number of cell = number of characters + 1
  - Array: several memory cells in a row.



Strings (“k”) vs. Characters (‘k’)

SEOUL NATIONAL UNIVERSITY

- ‘k’ is a basic type (char) but “k” is a derived type, an array of char
- “k” is actually consists of two chracters, ‘k’ and ‘\0’.

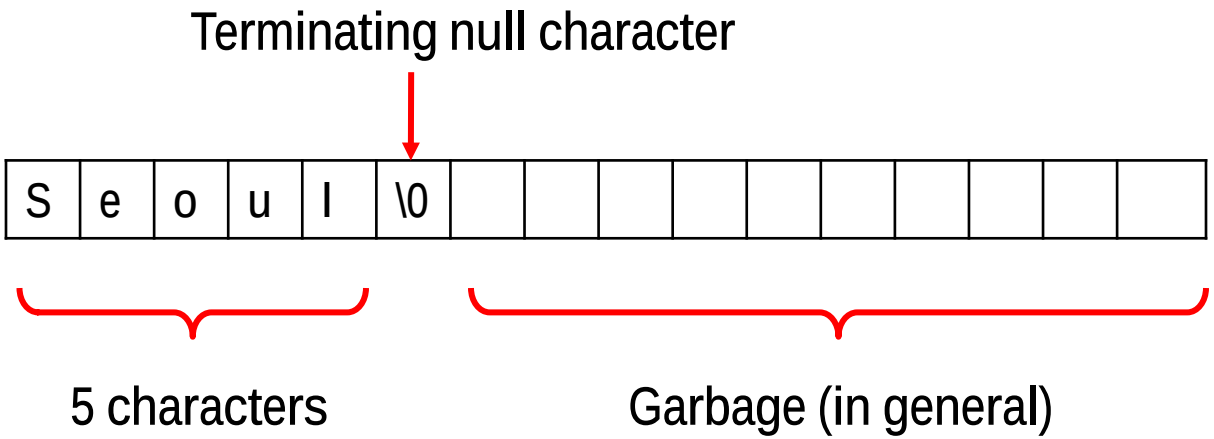
character ‘k’ 

string “k” 

strlen() function

SEOUL NATIONAL UNIVERSITY

- gives an exact number of characters in a string
- String-related functions - string.h header file
  - strlen()
- Standard input/output functions – stdio.h
  - printf(), scanf()



character string constant

SEOUL NATIONAL UNIVERSITY

- #define PRAISE “What a super marvelous name!”

Constants and the C preprocessor (C 전처리기)

SEOUL NATIONAL UNIVERSITY

- #define *NAME* value
  - NAME : symbolic name, usually in capital letters
  - No semicolon – just compile-time substitution
  - =

    ex) #define INTEREST 0.05

    …

    Payment = INTEREST * owed;
- Meaning of a value is clear
- Only need to alter the definition once – when it was used multiple times.

Manisfest constants (명단상수) and the C preprocessor (C 전처리기)

SEOUL NATIONAL UNIVERSITY

- #define can be used for characters and string constants
  - valid

    #define BEEP '\a'

    #define TEE 'T'

    #define OOPS "Now you have done it!"
  - Invalid

    #define TOES = 20

limits.h & float.h

SEOUL NATIONAL UNIVERSITY

- C header files limits.h and float.h supply detailed info about size limits of integer and float types

```c
// defines.c -- uses defined constants from limit.h and float.
#include <stdio.h>
#include <limits.h>    // integer limits
#include <float.h>     // floating-point limits
int main(void)
{
    printf("Some number limits for this system:₩n");
    printf("Biggest int: %d₩n", INT_MAX);
    printf("Smallest long long: %lld₩n", LLONG_MIN);
    printf("One byte = %d bits on this system.₩n", CHAR_BIT);
    printf("Largest double: %e₩n", DBL_MAX);
    printf("Smallest normal float: %e₩n", FLT_MIN);
    printf("float precision = %d digits₩n", FLT_DIG);
    printf("float epsilon = %e₩n", FLT_EPSILON);

    return 0;
}
```

```
C:₩Windows₩system32₩cmd.exe
Some number limits for this system:
Biggest int: 2147483647
Smallest long long: -9223372036854775808
One byte = 8 bits on this system.
Largest double: 1.797693e+308
Smallest normal float: 1.175494e-038
float precision = 6 digits
float epsilon = 1.192093e-007
계속하려면 아무 키나 누르십시오 . . .
```

printf(): an output function (출력함수)

SEOUL NATIONAL UNIVERSITY

- printf("Control-string", item1, item2, …);
  - ex) printf("The value of pi is %f.\n", PI);

Format specifier (포맷지정자)

- Format Specifier
  - %d signed decimal integer
  - %c single character
  - %f floating-point number, decimal notation
  - %e floating-point number, e-notation
  - %s character string
  - %u unsigned decimal integer
  - %% prints a percent sign

printf()
example

SEOUL NATIONAL UNIVERSITY

```c
/* printout.c -- 몇 가지 포맷 지정자를 사용한다 */
#include <stdio.h>
#define PI 3.141593
int main(void)
{
    int number = 5;
    float espresso = 13.5;
    int cost = 3100;

    printf("%d명의 CEO가 %f잔의 에스프레소를 마셨다.₩n", number,
        espresso);
    printf("The value of pi is %f.₩n", PI);
    printf("잘 가시라! 그대는 내가 소유하기에 과분하여라.₩n");
    printf("%c%d₩n", '$', 2 * cost);

    return 0;
}
```

```
5명의 CEO가 13.500000잔의 에스프레소를 마셨다.
The value of pi is 3.141593.
잘 가시라! 그대는 내가 소유하기에 과분하여라.
$6200
계속하려면 아무 키나 누르십시오 . . .
```

printf()
format modifiers

- Modify a basic conversion specification by inserting modifiers between the % and defining conversion character.
  - ex) printf (“ The value of pi is %6.2f.\n”, PI)
  - flag: -, +, space, #, 0
  - digits:The minimum field width (최소 필드 너비). A wider field will be used if the printed number won’t fit in the field. ex) %4d
  - .digits: Precision. Number of digit to the right of the decimal. ex) %5.2f
  - h indicates a **short** or **unsigned short int** value. ex) %hu
  - l indicates a long int or unsigned long int. ex) %ld

printf()
flag

- - item is left-justified. Ex) %-20d
- + signed values are displayed. +/-
- Space space for positive, - for negative
- # produce initial 0 for octal (8진 수) and 0x for hexadecimal (16진 수). ex) %#o
- 0 fill the field width with zeros. ex) %010d

printf()
example (1)

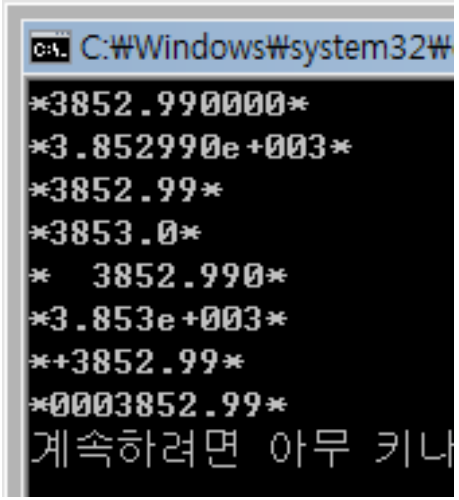
SEOUL NATIONAL UNIVERSITY

```c
// floats.c -- some floating-point combinations
#include <stdio.h>

int main(void)
{
    const double RENT = 3852.99;   // const-style constant

    printf("*%f*\n", RENT);
    printf("*%e*\n", RENT);
    printf("*%4.2f*\n", RENT);
    printf("*%3.1f*\n", RENT);
    printf("*%10.3f*\n", RENT);
    printf("*%10.3e*\n", RENT);
    printf("*%+4.2f*\n", RENT);
    printf("*%010.2f*\n", RENT);

    return 0;
}
```



- Default (%f):
  - field width (필드너비): whatever it takes
  - Number of digits to the right of decimal(소수점 아래자리수): 6
- Default (%e):
  - one digit to the left of decimal,
  - six digit to the right

printf()
example (2)

SEOUL NATIONAL UNIVERSITY

```c
/* flags.c -- illustrates some formatting flags */
#include <stdio.h>
int main(void)
{
    printf("%x %X %#x₩n", 31, 31, 31);
    printf("**%d**% d**% d**₩n", 42, 42, -42);
    printf("**%5d**%5.3d**%05d**%05.3d**₩n", 6, 6, 6, 6);

    return 0;
}
```

```
C:₩Windows₩system32₩cmd.exe
1f 1F 0x1f
**42** 42**-42**
**    6**   006**00006**   006**
계속하려면 아무 키나 누르십시오
```

```c
/* strings.c -- string formatting */
#include <stdio.h>
#define BLURB "Authentic imitation!"
int main(void)
{
    printf("/%2s/₩n", BLURB);
    printf("/%24s/₩n", BLURB);
    printf("/%24.5s/₩n", BLURB);
    printf("/%-24.5s/₩n", BLURB);

    return 0;
}
```

```
C:₩Windows₩system32₩cmd.exe
/Authentic imitation!/
/    Authentic imitation!/
/                   Authe/
/Authe                   /
계속하려면 아무 키나 누르십시오 . . .
```

Format Specifier

SEOUL NATIONAL UNIVERSITY

```c
/* floatcnv.c -- mismatched floating-point conversions */
#include <stdio.h>
int main(void)
{
    float n1 = 3.0;
    double n2 = 3.0;
    long n3 = 2000000000;
    long n4 = 1234567890;

    printf("%.1e %.1e %.1e %.1e\n", n1, n2, n3, n4);
    printf("%ld %ld\n", n3, n4);
    printf("%ld %ld %ld %ld\n", n1, n2, n3, n4);

    return 0;
}
```

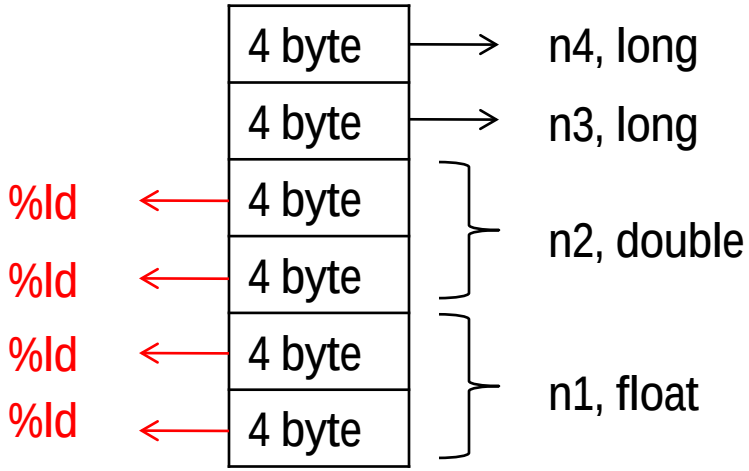
```
C:\Windows\system32\cmd.exe
3.0e+000 3.0e+000 3.1e+046 0.0e+000
2000000000 1234567890
0 1074266112 0 1074266112
계속하려면 아무 키나 누르십시오 . . .
```

- Original value is not replaced by format specifier!

printf()
mismatched conversion

SEOUL NATIONAL UNIVERSITY

```
float n1;
double n2;
long n3, n4;
...
printf("%ld %ld %ld %ld\n", n1, n2, n3, n4);
```



printf() removed values from stack as type **long**

Arguments n1 and n2 placed on stack as **double**, n3 and n4 as type **long**

printf()
return value

SEOUL NATIONAL UNIVERSITY

- a C function generally has a return value – a value that the function computes and returns to the calling program.
- printf() returns the number of characters it printed (자신이 출력한 문자의 수를 리턴한다).

```c
/* prntval.c -- finding printf()'s return value */
#include <stdio.h>
int main(void)
{
    int bph2o = 100;
    int rv;

    rv = printf("%d C is water's boiling point.\n", bph2o);
    printf("The printf() function printed %d characters.\n",
            rv);
    return 0;
}
```

1.Print information

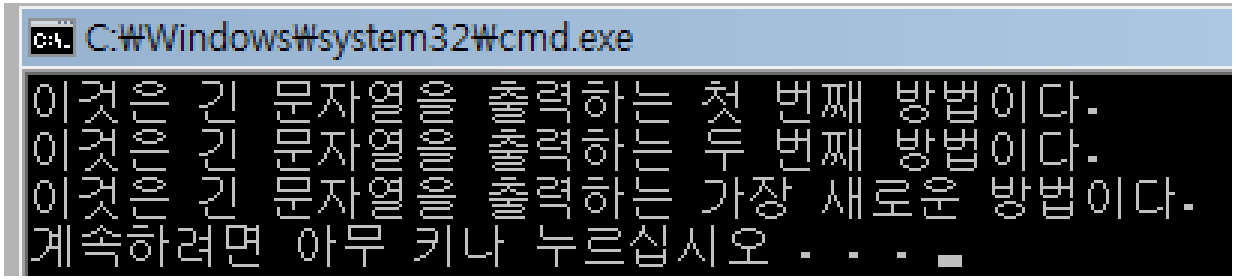
2.Assign a value

```
C:\Windows\system32\cmd.exe
100 C is water's boiling point.
The printf() function printed 32 characters.
계속하려면 아무 키나 누르십시오 . . .
```

printf()
printing long strings

SEOUL NATIONAL UNIVERSITY

```c
/* longstrg.c -- 긴 문자열의 출력 */
#include <stdio.h>
int main(void)
{
    printf("이것은 긴 문자열을 출력하는 ");
    printf("첫 번째 방법이다.₩n");
    printf("이것은 긴 문자열을 출력하는 ₩
두 번째 방법이다.₩n");
    printf("이것은 긴 문자열을 출력하는 "
        "가장 새로운 방법이다.₩n");      /* ANSI C */
    return 0;
}
```



scanf()

SEOUL NATIONAL UNIVERSITY

- **scanf()** function convert string inputs into various forms: integers, floating-point numbers, characters, strings.
- Inverse of **printf()**: **printf()**와 반대
- scanf("Control-string", &item1, &item2, …);

포멧문자열

전달인자

전달인자의 주소,
printf()와 차이점

  - ex) scanf("%d", &pet);

scanf()
argument list

SEOUL NATIONAL UNIVERSITY

```c
// input.c -- &를 언제 사용하는가
#include <stdio.h>
int main(void)
{
    int age;            // 변수
    float assets;       // 변수
    char pet[30];       // 문자열

    printf("나이, 재산, 좋아하는 애완동물을 입력하시오.₩n");
    scanf("%d %f", &age, &assets); // 여기에는 &를 사용한다
    scanf("%s", pet);              // 문자 배열에는 &를 사용하지 않는다
    printf("%d $%.2f %s₩n", age, assets, pet);

    return 0;
}
```

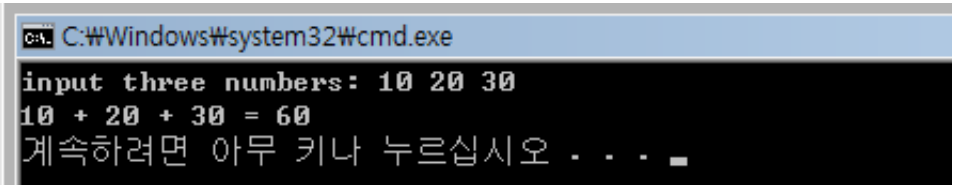
```
나이, 재산, 좋아하는 애완동물을 입력하시오.
30
10000
dog
30 $10000.00 dog
계속하려면 아무 키나 누르십시오 . . .
```

- For basic variables (int, float, char): use &
- For string: Don't use &
- & denotes the address of the variables and strings are already address.

scanf()
input

SEOUL NATIONAL UNIVERSITY

```c
#include<stdio.h>
int main(void)
{
    int num1, num2, num3, total;
    printf("input three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);
    total = num1 + num2 + num3;
    printf("%d + %d + %d = %d\n", num1, num2, num3, total);
}
```



- scanf() function uses whitespace (구분자) to divide the input into separate fields;
  - Enter
  - Tabs
  - Spaces
- Exception: %c

scanf()
format specifier

SEOUL NATIONAL UNIVERSITY

- Similar to printf()
- %d decimal integer
- %c character
- %e, %f floating-point numbers
- %s string. From non-whitespace character to whitespace character. ex) Tiger ~~Woods~~
  To read single character: getchar() function
  To read a string with space: gets() function.

scanf()
format specifier

SEOUL NATIONAL UNIVERSITY

```c
#include <stdio.h>
int main(void)
{
    int num;
    char ch,str[20];

    printf("input number, character, string: ");
    scanf("%d%c %s",&num, &ch, str);
    printf("number: %d\t character: %c\t string: %s\n", num, ch, str);

    return 0;
}
```

```
C:\Windows\system32\cmd.exe
input number, character, string: 50 k seoul
number: 50       character:      string: k
계속하려면 아무 키나 누르십시오 . . .
```

- %c : space is not skipped
  - With ‘space’, skip to non-whitespace character
- Valid form: %d %c %s

space

scanf()
return value, * modifier

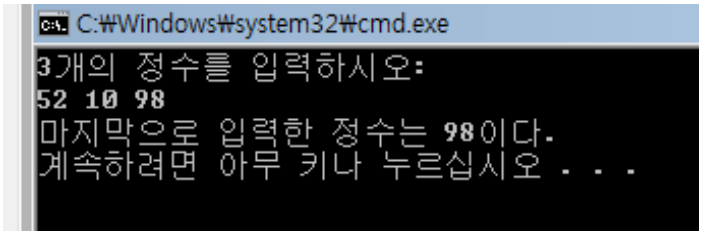
- Scanf() returns the number of items that it successfully reads (성공적으로 읽은 항목의 수) – printf()???

- * causes the function to skip over corresponding input.

```c
#include <stdio.h>
int main(void)
{
    int n;

    printf("3개의 정수를 입력하시오:₩n");
    scanf("%*d %*d %d", &n);
    printf("마지막으로 입력한 정수는 %d이다.₩n", n);

    return 0;
}
```



scanf()
format modifier

SEOUL NATIONAL UNIVERSITY

- * suppress assignment (읽지 않고 넘어감)
  ex) “%*d”
- Digits Maximum field width. Stops at the first whitespace character. ex) “%10s”

Summary

SEOUL NATIONAL UNIVERSITY

- Character string
- printf()
  - Format specifier, Format modifier
- scanf()
  - Format specifier, Format modifier
- Miscellaneous
  - Other header files
  - Manifest constant (명단함수)

Next Lecture
Chapter 5. C primer Plus

SEOUL NATIONAL UNIVERSITY

- Operator (연산자): + - = * / ++ --

- While loop

```
While (height < 170)          // start of a loop
  {                           // start of a block
        ....
  }                           / end of a block
```

- There will be another homework next week.