

Homework #1

C Programming

Mission



- Solve problem 1~3
- Submission
 - Submit a zip file containing hw2_1.c, hw2_2.c, and hw2_3.c on HISNet.
 - Source files hw2_*.c should contain the followings:
 - Algorithm in pseudo code as comments preceded by “//”.
 - See the next page.
 - C program code that implements the pseudo code
- cf. Upload only source files (.c files)
 - Search for *.c files under the project directory and copy them to hw2_1.c, hw2_2.c, or hw2_2.c.
- Due date: PM 11:00, Sep. 17th

Code Example (add.c)



```
#include <stdio.h>

int main()
{
    int a = 0, b = 0;
    int sum = 0;

    // read two integer numbers
    printf("Input two numbers: ");
    scanf("%d %d", &a, &b);

    // add the two numbers
    sum = a + b;

    // print the result
    printf("%d + %d = %d\n", a, b, sum);

    return 0;
}
```

Honor Code Guidelines (Korean)

■ “과제”

- 과제는 교과과정의 내용을 소화하여 실질적인 활용 능력을 갖추기 위한 교육활동이다. 학생은 모든 과제를 정직하고 성실하게 수행함으로써 과제에 의도된 지식과 기술을 얻기 위해 최선을 다해야 한다.
- 담당교수가 명시적으로 허락한 경우를 제외하고 다른 사람이 작성하였거나 인터넷 등에서 획득한 과제물, 또는 프로그램 코드의 일부, 또는 전체를 이용하는 것은 부정행위에 해당한다.
- 자신의 과제물을 타인에게 보여주거나 빌려주는 것은 공정한 평가를 방해하고, 해당 학생의 학업 성취를 저해하는 부정행위에 해당한다.
- 팀 과제가 아닌 경우 두 명 이상이 함께 과제를 수행하여 이를 개별적으로 제출하는 것은 부정행위에 해당한다.
- 서로 다른 학생이 제출한 제출물간 유사도가 통상적으로 발생할 수 있는 정도를 크게 넘어서는 경우, 또는 자신이 제출한 과제물에 대하여 구체적인 설명을 하지 못하는 경우에는 부정행위로 의심받거나 판정될 수 있다.

Problem 0



- Memorize the types and conversion specs on the next page
 - We could have a pop quiz about the following topics
 - Types
 - Variable declarations
 - Conversion specifications for formatted I/O

Types and Conversion Specifications

■ Formatted input: `scanf`

■ Integer

```
int i;  
scanf("%d", &i);
```

■ Float

```
float f;  
scanf("%f", &f);
```

■ Character

```
char ch;  
scanf("%c", &ch);
```

■ String

```
char str[100];  
scanf("%s", str);
```

■ Formatted output: `printf`

■ Integer

```
printf("value = %d\n", i);
```

■ Float

```
printf("value = %f\n", f);
```

■ Character

```
printf("ch = %c\n", ch);
```

■ String

```
printf("str = %s\n", str);
```

*%d, %f, %c, %s are **conversion specifications***

Problem 1 (hw2_1.c)



- Read three characters and print them in ascending order

Ex) Input three characters: **a c d**

a-c-d

Ex) Input three characters: **c d a**

a-c-d

Ex) Input three characters: **a d c**

a-c-d

- Hint: if-statement

```
if(<condition>){    // ex) x % 2 == 0, x < y
    <statement1>    // executes when <condition> is true
} else {
    <statement2>    // executes when <condition> is false
}
```

Problem 1



■ Partial algorithm

```
char c1 = 0, c2 = 0, c3 = 0;
```

```
read three characters
```

```
put space character between %c's in scanf(), ("%c %c %c")
```

```
if c1 precedes c2 {  
    if c2 precedes c3 {  
        display c1, c2, c3  
    } else {  
        if c1 precedes c3  
            display c1, c3, c2  
        else  
            display c3 c1, c2  
    }  
} else {  
    // build an algorithm similar to the above  
}
```


Problem 2 (hw2_2.c)



- Read the full name and gender of the user and print greeting message.

Ex)

Input your full name (first middle last): Injung Eric Kim

Input your gender (m or f): m

You name is Injung E. Kim

Nice to see you Mr. Kim. // for female, display Ms.

- Hint

- i^{th} character of a string s can be retrieved by $s[i]$

Ex) `char str[32] = "hello";`

`printf("%c", str[0]);` // displays 'h'

Problem 3 (hw2_3.c)



- Read five positive integers and print the maximum.

Ex)

Input an integer: 10

Input an integer: 30

Input an integer: 40

Input an integer: 20

Input an integer: 10

maximum = 40

- Hint: repeating five times

```
int i = 0;           // loop variable
for(i = 0; i < 5; i++){
    <statement>     // eg. printf("i = %d\n", i);
}
```

Problem 3



■ Algorithm

Initialize *max* by zero

Repeat 5 times

 Read an integer *x*

 If *x* is greater than *max*, update *max* with *x*

Print *max*