

# Homework #4

C Programming

# Mission



- Solve problem 1~3
- Submission
  - Submit a zip file containing hw4\_1.c, hw4\_2.c, and hw4\_3.c on HISNet.
  - Source files hw4\_\*.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 hw4\_1.c, hw4\_2.c, or hw4\_2.c.
    - The code should be properly indented.
- Due date: PM 11:00, Oct. 14<sup>th</sup>

# 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;
}
```

# Indentation is Crucial for Readability



## ■ Properly indented code

```
int main()
{
    int height = 0;
    int i = 0, j = 0;

    do {
        printf("Input the height of triangle: ");
        scanf("%d", &height);
    } while (height % 2 == 0);

    for(i = 1; i <= height; i++){
        int start = height - i;
        int end = height + i - 1;

        for(j = 0; j < start; j++)
            putchar(' ');
        for(; j < end; j++)
            putchar('*');
        putchar('\n');
    }

    system("PAUSE");

    return 0;
}
```

## ■ Not indented code

```
int main()
{
    int height = 0;
    int i = 0, j = 0;

    do {
        printf("Input the height of triangle: ");
        scanf("%d", &height);
    } while (height % 2 == 0);

    for(i = 1; i <= height; i++){
    int start = height - i;
    int end = height + i - 1;

    for(j = 0; j < start; j++)
    putchar(' ');
    for(; j < end; j++)
    putchar('*');
    putchar('\n');
    }

    system("PAUSE");

    return 0;
}
```

# Rules of Indentation

- Start function header from the first column
- Use tab to represent indentation level.
- The body of functions, blocks, selection (if, switch), or repetition (while, for, do-while) statements should be indented one more level.

Ex) if(x % 2 == 0)

↔ printf("x is an even number.\n");

Ex) do {

↔ printf("Input a positive number:");

↔ scanf("%d", &x);

} while(x <= 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 following code

```
#include <stdio.h>
```

```
float GetCircleSize(float radius); // function declaration
```

```
int main()  
{
```

```
    float r = 0.F;
```

```
    printf("Input radius of a circle : ");  
    scanf("%f", &r);
```

```
    printf("The size of circle is %f\n", GetCircleSize(r)); // function call
```

```
    return 0;  
}
```

```
// function definition
```

```
float GetCircleSize(float radius)
```

```
{
```

```
    float size = radius * radius * 3.141592F;
```

```
    return size;
```

```
}
```

# Problem 1



- Read a number N and print the sum of numbers from 1 to N.

- Repeat until the user inputs EOF. (CTRL-Z or CTRL-D)  
Ex)

Input a number N: 3

Sum of numbers from 1 to N is 6

Input a number N: 8

Sum of numbers from 1 to N is 36

Input a number N: 10

Sum of numbers from 1 to N is 55

Input a number N: 28

Sum of numbers from 1 to N is 406

Input a number N: ^Z

// EOF

Bye!



# Problem 1

---



## ■ Requirements

- Define and use a function “int Accumulate(int n);” that receives a number  $n$  as the parameter and returns the sum of numbers from 1 to  $n$ .
- Declare Accumulate() before main().
- Use Accumulate() in main to solve the problem.

# Problem 2



## ■ Read numbers and draw histogram.

- Write a function “void DrawBar(int length, char c);” that draws a bar.
    - *length*: the length of the bar
    - *c*: the character to draw with
- ex) “DrawBar(10, ‘A’);” should print “AAAAAAAAAA”

Ex)

Input numbers (–999) to stop: 10 25 40 20 30 –999

#####

#####

#####

#####

#####

- Hint: Run the following program and type “10 25 40 20 30 –999” as the input.

# Problem 2

---



```
#include <stdio.h>

int main()
{
    int i = 0;
    int x = 0;

    printf("Input numbers: ");
    for(i = 0; x != -999; i++){
        scanf("%d", &x);
        if(x != -999)
            printf("%d-th number is %d\n", i, x);
    }

    return 0;
}
```

# Problem 3



- Write a function “void CopyString(char dest[], char src[]);” that copies a string *src* to *dest*.

- Combined with the main() on the next page, the output should be as follows:

Input a text line: **God bless you!**

text = [God bless you!]

text2 = []

Calling CopyString(text2, text);     // copy text to text2

text = [God bless you!]

text2 = [**God bless you!**]

# Problem 3

---



## ■ Main()

```
int main()
{
    char text[256] = "";
    char text2[256] = "";

    printf("Input a text line: ");
    gets(text);

    printf("text = [%s]\n", text);
    printf("text2 = [%s]\n", text2);

    printf("Calling CopyString(text2, text);\n");
    CopyString(text2, text);

    printf("text = [%s]\n", text);
    printf("text2 = [%s]\n", text2);

    return 0;
}
```