

Homework #6

C Programming

Mission



- Write a puzzle game by solving problem 1~3
- Submission
 - Submit a zip file containing hw6_1.c, hw6_2.c, and hw6_3.c on HISNet.
 - Source files hw6_*.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 hw6_1.c, hw6_2.c, or hw6_2.c.
 - The code should be properly indented.
- Due date: PM 11:00, Nov. 26th

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 1



■ Write the following two functions:

// use constants WIDTH and HEIGHT to implement the functions

```
#define WIDTH 5
```

```
#define HEIGHT 4
```

```
void FillNumPuzzle(int puzzle[][WIDTH]);
```

- Fill puzzle with numbers from 0 to (WIDTH * HEIGHT - 1)
- **Zero represents the hole.**

```
void DisplayNumPuzzle(int puzzle[][WIDTH]);
```

- Display all numbers in puzzle.
- **Print blank spaces for zero.**

Problem 1



■ main()

```
int main()
{
    int puzzle[HEIGHT][WIDTH];

    FillNumPuzzle(puzzle);
    DisplayNumPuzzle(puzzle);

    return 0;
}
```

■ Result

```
    1   2   3   4
  5   6   7   8   9
10  11  12  13  14
15  16  17  18  19
```

// zero was not printed

Problem 2



- Extend the solution of Problem 1 by adding the following two functions

`void ShuffleNumPuzzle(int puzzle[][WIDTH])`

- Shuffle all numbers in puzzle.

`void GetZeroPosition(int puzzle[][WIDTH], int *r, int *c)`

- Find the coordinate of zero. Put the vertical coordinate in *r and the horizontal coordinate in *c.
- See the example code and result in the next pages.

Problem 2



■ main()

```
int main()
{
    int puzzle[HEIGHT][WIDTH];
    int i = 0;

    srand(time(NULL));
    FillNumPuzzle(puzzle);
    DisplayNumPuzzle(puzzle);

    for(i = 0; i < 3; i++){
        int hole_r = 0, hole_c = 0;

        ShuffleNumPuzzle(puzzle);
        GetZeroPosition(puzzle, &hole_r, &hole_c);
        printf("Hole position: row = %d, col = %d\n", hole_r, hole_c);
        DisplayNumPuzzle(puzzle);
    }

    return 0;
}
```

Problem 2



■ Result

```

    1  2  3  4
  5  6  7  8  9
10 11 12 13 14
15 16 17 18 19
Hole position: row = 2, col = 0
  3  8 14 10  9
19 15 13  7 12
   4 17  6 11
18 16  5  1  2
Hole position: row = 2, col = 3
14  7 15  9 11
  5 12  2  4 10
16 18  8     6
17  3  1 13 19
Hole position: row = 3, col = 2
  1  4  9  3 10
14  5  7 13  2
  8 11 12 18  6
17 19     15 16
```

Problem 2



■ Algorithm for ShuffleNumPuzzle()

Repeat on i from 0 to $\text{HEIGHT} - 1$

Repeat on j from 0 to $\text{WIDTH} - 1$

Randomly select $i2$ in $[0, \text{HEIGHT})$ and $j2$ in $[0, \text{WIDTH})$

Swap $\text{puzzle}[i][j]$ and $\text{puzzle}[i2][j2]$

Problem 3



■ Complete the puzzle game

- Read the partial solution in hw6_3_skeleton.c carefully.
- Then, complete main() by writing up and down movement.
- Use the four functions in Problem 1 and 2.
- Use Console.c and Console.h

■ Compilation

- On Windows

```
gcc hw6_3.c Console.c -D_WINDOWS
```
- On Mac

```
gcc hw6_3.c Console.c -D_MAC
```