

Review #01

이름: _____

학번: _____

(제출 시 .vs 폴더, debug 폴더를 지운 다음 .zip 으로 압축 후 제출)

1. 아래의 요구사항을 만족하는 프로그램을 만드시오. ([시작코드]는 변경하지 말 것)

[요구사항]

- get_command(): 아래의 메뉴를 화면에 출력하고 키보드로부터 command 를 입력 받음

```
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>
```

- get_size(): "Enter the size in (size x size): "를 화면에 출력하고 키보드로부터 size 를 입력받음

- gen_matrix(): 정방형(size x size) 행렬을 생성하고 rand()를 통해 각 element 에 [0, 100]범위의 값을 저장

- sort_matrix_row(): 행렬의 각 행을 오름차순으로 정렬 후 "Completed"를 화면에 출력

1) 이때, 각 행에 대한 정렬은 sort_array()를 사용(각 행을 array 로 입력)

2) sort_array()는 Bubble sort 방법으로 정렬, 각 element 의 값 교환을 위해 swap()사용

3) Bubble sort: 0 번째 index 부터 마지막 index 까지 이웃 원소간 비교를 통해 작은값은 왼쪽, 큰값은 오른쪽 위치로 교환시키는 방법

오름차순으로 정렬하는 거품정렬의 과정은 다음과 같다.

예제:

```
55 07 78 12 42 초기값[파란색은 sorting]
07 55 78 12 42 첫 번째 패스(pass)
07 55 78 12 42
07 55 12 78 42
07 55 12 42 78 두 번째 패스(pass)
07 55 12 42 78
07 12 55 42 78
07 12 42 55 78 세 번째 패스(pass)
07 12 42 55 78 네 번째 패스(pass)
07 12 42 55 78 다섯 번째 패스(pass)
07 12 42 55 78 정렬 끝
```

```
procedure bubbleSort( A : list of sortable items ) defined as
  for each i in 1 to length(A) do:
    for each j in length(A) downto i + 1 do:
      if A[ j ] < A[ j - 1 ] then
        swap( A[ j ], A[ j - 1 ] )
      end if
    end for
  end for
end procedure
```

Pseudo-code:

- print_matrix(): 화면에 렬의 각 원소를 출력(cout<< setw(4) << matrix[row][col]을 이용할 것)

- save_matrix(): "matrix.txt"에 아래와 같이 저장 (첫 줄: size, 다음 줄: 원소들)후 화면에 "Saved"를 출력



```

3
41 72 85
38 69 80
65 68 96

```

- free_matrix(): 행렬 할당 해제

[시작코드]

```

#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>
using namespace std;

void get_command(/*작성할 것*/);// 10점
void get_size(/*작성할 것*/);// 10점
int** gen_matrix(/*작성할 것*/);// 20점
void swap(int* a, int* b);// 10점
void sort_array(int* ary, int size);// 10점
void sort_matrix_row(/*작성할 것*/);// 10점
void print_matrix(/*작성할 것*/);// 10점
void save_matrix(/*작성할 것*/);//10점
void free_matrix(/*작성할 것*/);// 10점

int main() {
    string command;
    int** matrix = NULL;
    int size = 0;
    while (1) {
        get_command(command);
        if(command == "1"){
            get_size(size);
            matrix = gen_matrix(size);
        }else if (command == "2") {
            print_matrix(matrix, size);
        }else if (command == "3") {
            sort_matrix_row(matrix, size);
        }else if (command == "4") {
            save_matrix(matrix, size);
        }else if (command == "0") {
            free_matrix(matrix, size);
            cout << "Exit the program.." << endl;
            exit(104);
        }else {
            cout << "Wrong command" << endl << endl;
        }
    }

    return 0;
}

```

[실행 화면]

```
C:\WINDOWS\system32\cmd.exe
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>1
Enter the size in (size x size): 3
```

```
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>2
 41 85 72
 38 80 69
 65 68 96
```

```
C:\WINDOWS\system32\cmd.exe
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>3
Completed
```

```
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>2
 41 72 85
 38 69 80
 65 68 96
```

```
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>4
Saved
```

```
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>5
Wrong command
```

```
1. Generate matrix
2. Print matrix
3. Sort matrix
4. Save matrix
0. Exit program
>>0
Exit the program..
```

계속하려면 아무 키나 누르십시오 . . . █