MEDİPOL UNV ISTANBUL MEDIPOL UNIVERSITY

Programming for Engineers

Fall 21-22

Assignment 2

Student Name: Gülsüm İkbal Avşar

Student ID: 64180006

Department: CoE/ EEE

Problem-1

Given any two-dimensional array **A** with n-rows and m-columns, where the values of n and m are specified by the user at run-time. The entries in **A** are integers between 1 and 100. Arrange the entries in array **A** according to the following policy:

- the small values must appear in the first (n / 2) rows in ascending order (notice that this is an integer division).
- the remaining values must appear in the remaining rows in descending order.

The following example shows how values in a 5 x 6 array are ordered.
(5 / 2) = 2, so the first two rows contain the small values in ascending order and the remaining 3 rows contain the remaining values in descending order.

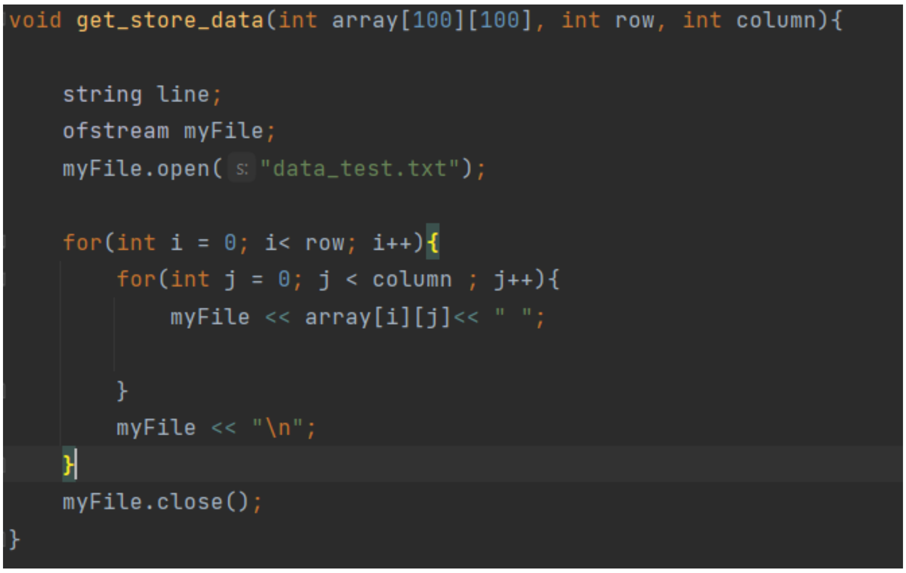
2	4	5	7	8	1
15	3	29	78	11	10
34	26	37	17	16	68
24	19	31	83	54	50
72	14	47	41	93	81

1	2	3	4	5	7
8	10	11	14	15	16
93	83	81	78	72	68
54	50	47	41	37	34
31	29	26	24	19	17

Get the size of the array (number of rows and columns) in the *main* function. Be careful to use appropriate call by value or call by reference in each of the following functions. Call all functions from inside *main*:

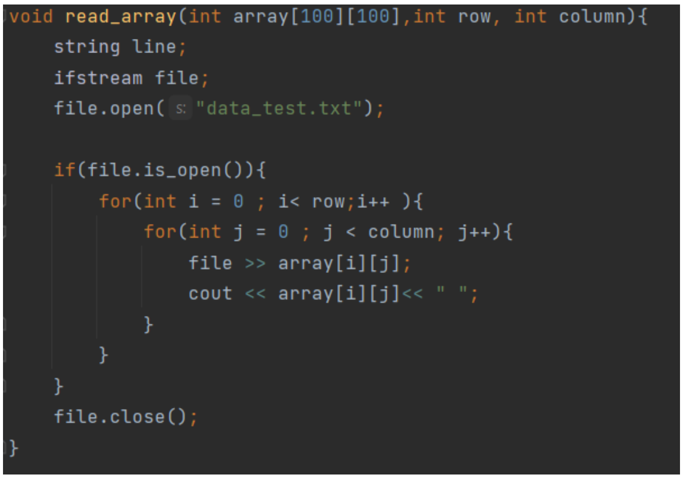
1. Write a function to get the data from the user and store into a file
2. Read the values from the file into the array
3. Write a function to sort in ascending order
4. Write a function to sort in descending order
5. Write a function to print any part of an array from row a to row b and from column r to column s. You must check and enforce that a<=b and r<=s and within the given dimensions of the array.

The Program Code



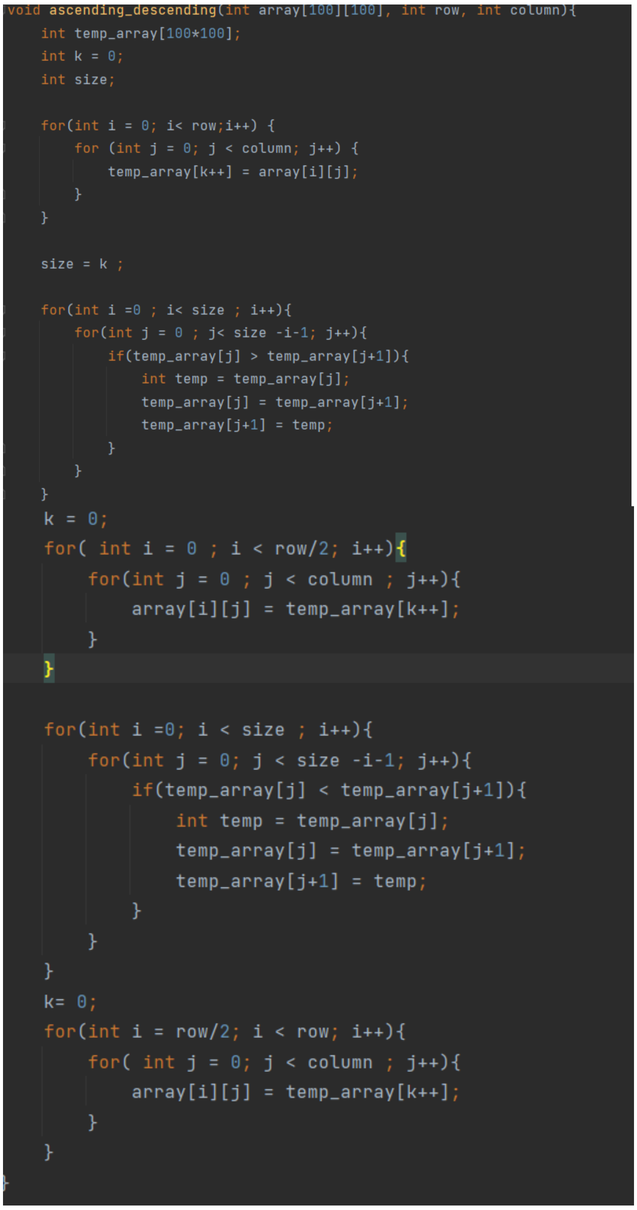
```
void get_store_data(int array[100][100], int row, int column){

    string line;
    ofstream myFile;
    myFile.open( s: "data_test.txt");

    for(int i = 0; i< row; i++){
        for(int j = 0; j < column ; j++){
            myFile << array[i][j]<< " ";

        }
        myFile << "\n";
    }
    myFile.close();
}
```

Figure-1 / Writing Data to File



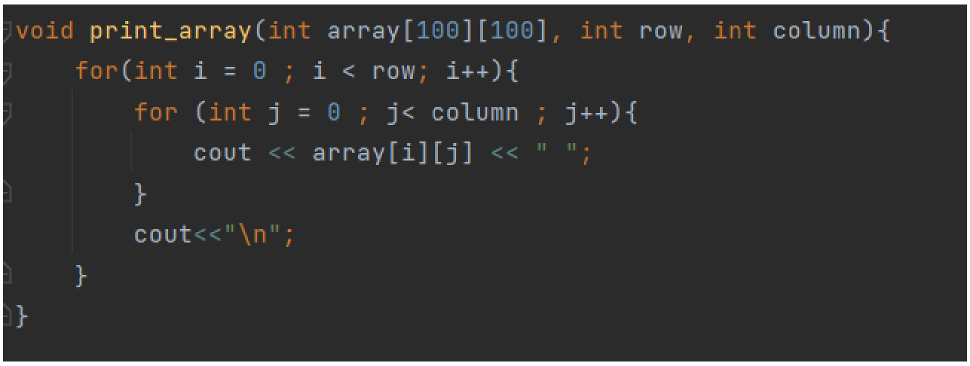
```
void read_array(int array[100][100],int row, int column){
    string line;
    ifstream file;
    file.open( s: "data_test.txt");

    if(file.is_open()){
        for(int i = 0 ; i< row;i++ ){
            for(int j = 0 ; j < column; j++){
                file >> array[i][j];
                cout << array[i][j]<< " ";
            }
        }
    }
    file.close();
}
```

Figure-2/ Reading Data from a File



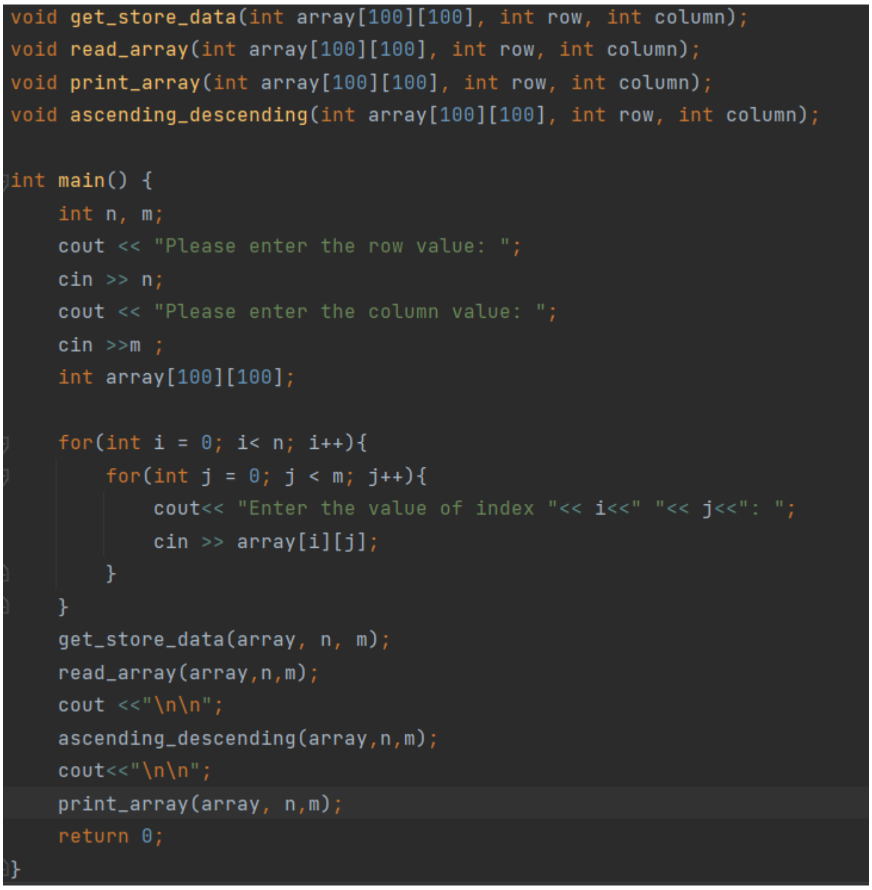
```
void ascending_descending(int array[100][100], int row, int column){
    int temp_array[100*100];
    int k = 0;
    int size;

    for(int i = 0; i< row;i++) {
        for (int j = 0; j < column; j++) {
            temp_array[k++] = array[i][j];
        }
    }

    size = k ;

    for(int i =0 ; i< size ; i++){
        for(int j = 0 ; j< size -i-1; j++){
            if(temp_array[j] > temp_array[j+1]){
                int temp = temp_array[j];
                temp_array[j] = temp_array[j+1];
                temp_array[j+1] = temp;
            }
        }
    }
    k = 0;
    for( int i = 0 ; i < row/2; i++){
        for(int j = 0 ; j < column ; j++){
            array[i][j] = temp_array[k++];
        }
    }

    for(int i =0; i < size ; i++){
        for(int j = 0; j < size -i-1; j++){
            if(temp_array[j] < temp_array[j+1]){
                int temp = temp_array[j];
                temp_array[j] = temp_array[j+1];
                temp_array[j+1] = temp;
            }
        }
    }
    k= 0;
    for(int i = row/2; i < row; i++){
        for( int j = 0; j < column ; j++){
            array[i][j] = temp_array[k++];
        }
    }
}
```

Figure-3/ Ordering Array Properly



```
void print_array(int array[100][100], int row, int column){
    for(int i = 0 ; i < row; i++){
        for (int j = 0 ; j< column ; j++){
            cout << array[i][j] << " ";
        }
        cout<<"\n";
    }
}
```

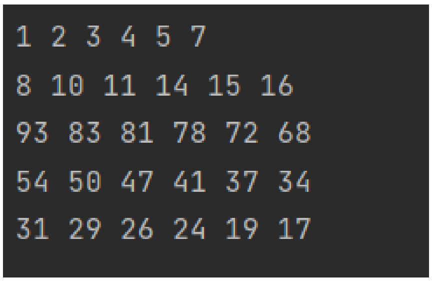
Figure-4/ Printing Array



```
void get_store_data(int array[100][100], int row, int column);
void read_array(int array[100][100], int row, int column);
void print_array(int array[100][100], int row, int column);
void ascending_descending(int array[100][100], int row, int column);

int main() {
    int n, m;
    cout << "Please enter the row value: ";
    cin >> n;
    cout << "Please enter the column value: ";
    cin >>m ;
    int array[100][100];

    for(int i = 0; i< n; i++){
        for(int j = 0; j < m; j++){
            cout<< "Enter the value of index "<< i<<" "<< j<<": ";
            cin >> array[i][j];
        }
    }
    get_store_data(array, n, m);
    read_array(array,n,m);
    cout <<"\n\n";
    ascending_descending(array,n,m);
    cout<<"\n\n";
    print_array(array, n,m);
    return 0;
}
```

Figure-5/ Main Function

Program Output



```
1 2 3 4 5 7
8 10 11 14 15 16
93 83 81 78 72 68
54 50 47 41 37 34
31 29 26 24 19 17
```

Figure-6/ Output

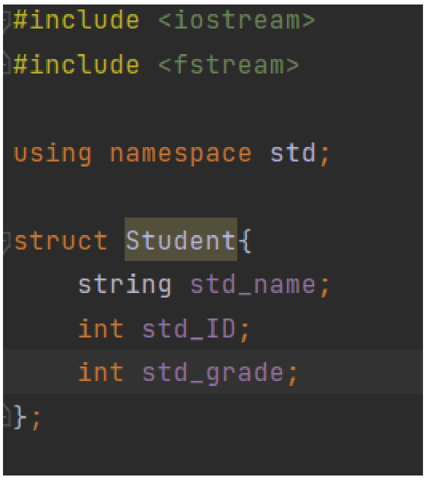
Problem-2

Define a *Student* structure that contains fields for the *student name*, *ID number* and *course grade* ($0 \leq grade \leq 100$). Then define *CourseGrades* array of *Student* structures that keeps track of the list of students, IDs and their final grades for a given course. Implement the following functions:

- Get the number of students in the course, their names, IDs and grades from the user
- Print the list of students, IDs and grades into a file
- Compute and display the average grade of all students
- Display the letter grades of students on the screen. Letter grades are assigned to students as follows:

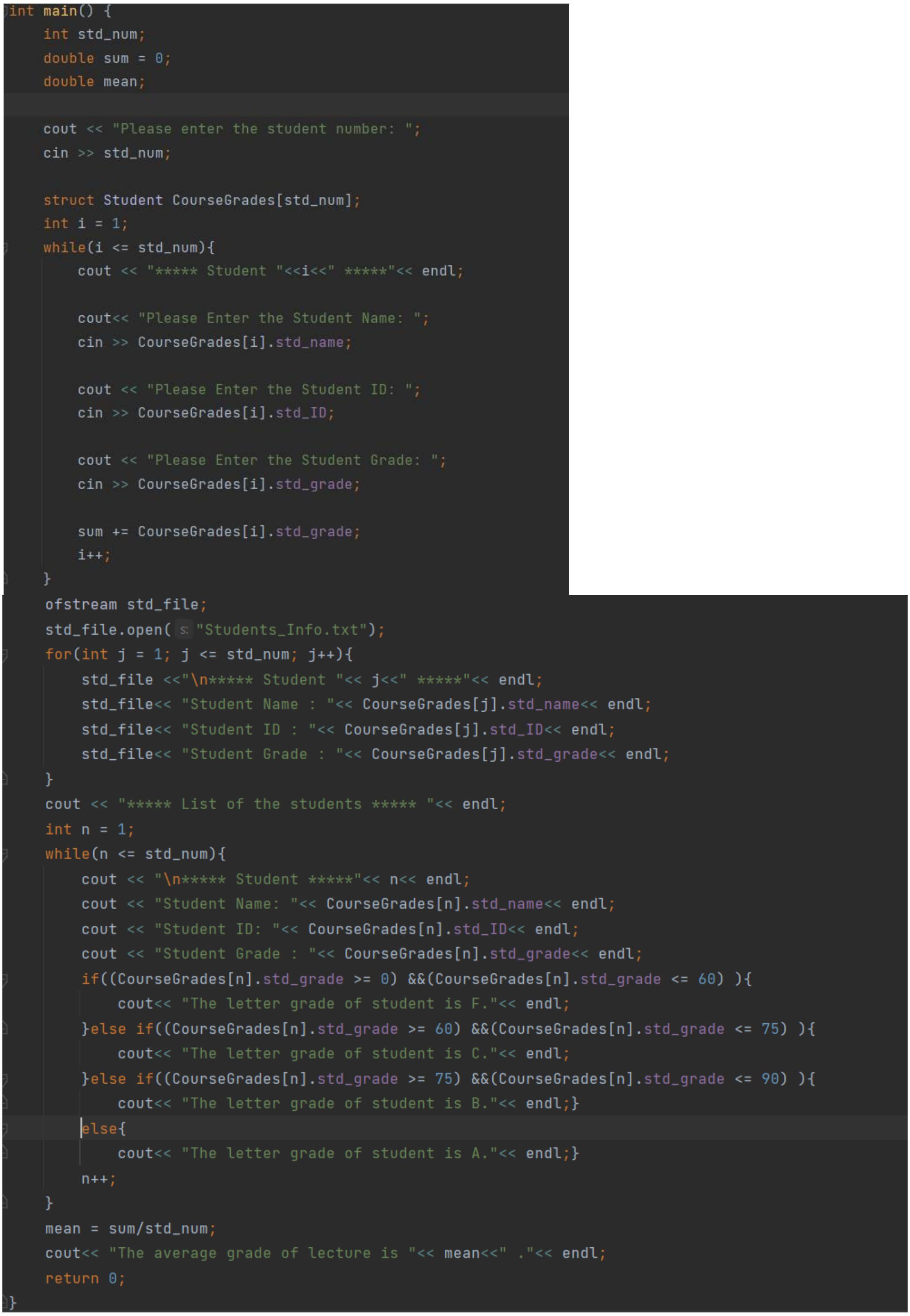
  $0 \leq grade < 60$ → F
  $60 \leq grade < 75$ → C
  $75 \leq grade < 90$ → B
  $90 \leq grade \leq 100$ → A

Program Code



```
#include <iostream>
#include <fstream>

using namespace std;

struct Student{
    string std_name;
    int std_ID;
    int std_grade;
};
```

Figure-7 / Student Structure



```cpp
int main() {
    int std_num;
    double sum = 0;
    double mean;

    cout << "Please enter the student number: ";
    cin >> std_num;

    struct Student CourseGrades[std_num];
    int i = 1;
    while(i <= std_num){
        cout << "***** Student "<<i<<" *****"<< endl;

        cout<< "Please Enter the Student Name: ";
        cin >> CourseGrades[i].std_name;

        cout << "Please Enter the Student ID: ";
        cin >> CourseGrades[i].std_ID;

        cout << "Please Enter the Student Grade: ";
        cin >> CourseGrades[i].std_grade;

        sum += CourseGrades[i].std_grade;
        i++;
    }
    ofstream std_file;
    std_file.open( s: "Students_Info.txt");
    for(int j = 1; j <= std_num; j++){
        std_file <<"\n***** Student "<< j<<" *****"<< endl;
        std_file<< "Student Name : "<< CourseGrades[j].std_name<< endl;
        std_file<< "Student ID : "<< CourseGrades[j].std_ID<< endl;
        std_file<< "Student Grade : "<< CourseGrades[j].std_grade<< endl;
    }
    cout << "***** List of the students ***** "<< endl;
    int n = 1;
    while(n <= std_num){
        cout << "\n***** Student *****"<< n<< endl;
        cout << "Student Name: "<< CourseGrades[n].std_name<< endl;
        cout << "Student ID: "<< CourseGrades[n].std_ID<< endl;
        cout << "Student Grade : "<< CourseGrades[n].std_grade<< endl;
        if((CourseGrades[n].std_grade >= 0) &&(CourseGrades[n].std_grade <= 60) ){
            cout<< "The letter grade of student is F."<< endl;
        }else if((CourseGrades[n].std_grade >= 60) &&(CourseGrades[n].std_grade <= 75) ){
            cout<< "The letter grade of student is C."<< endl;
        }else if((CourseGrades[n].std_grade >= 75) &&(CourseGrades[n].std_grade <= 90) ){
            cout<< "The letter grade of student is B."<< endl;}
        else{
            cout<< "The letter grade of student is A."<< endl;}
        n++;
    }
    mean = sum/std_num;
    cout<< "The average grade of lecture is "<< mean<<" ."<< endl;
    return 0;
}
```

Figure-8/ Main Function