The British University in Egypt **Faculty of Informatics and Computer Science** **Programming Section**	The BRITISH UNIVERSITY IN EGYPT
Module Name: Introduction Programming	**Module code:** 22CSCI01P
File Type: Revision on 1. Functions 2. Files 3. Arrays	**Outcomes:** 1. Tracing 2. Programming

1. Tracing a piece of code:

1) What is the output of the following program?

```
#include <iostream>
using namespace std;
struct Pixel{
    int C, R;
};
void Display(Pixel P){
    cout << "Col " << P.C << " Row " << P.R << ", " ;
}
int main(){
    Pixel X = { 40,50 }, Y, Z;
    Z = X;
    X.C += 10;
    Y = Z;
    Y.C += 10;
    Y.R += 20;
    Z.C -= 15;
    Display(X);
    Display(Y);
    Display(Z);

    return 0;
}
```

2) What is the output of the following program?

```
#include <iostream>
using namespace std;
struct Play{
    int score, bonus;
};
void calculate(Play& P, int N = 10){
    P.score++;
    P.bonus += N;
}
int main(){
    Play PL = { 10, 15 };
    calculate(PL, 5);
    cout << PL.score << ":" << PL.bonus << ", ";
    calculate(PL);
    cout << PL.score << ":" << PL.bonus << ", ";
    calculate(PL, 15);
    cout << PL.score << ":" << PL.bonus << ", ";

    return 0;
}
```

3) What is the output of the following program?

```
#include <iostream>
using namespace std;
struct MyBox{
    int length, breadth, height;
};
void dimension(MyBox M){
    cout << M.length << "x" << M.breadth << "x";
    cout << M.height << "\t";
}
int main(){
    MyBox B1 = { 10, 15, 5 }, B2, B3;
    ++B1.height;
    dimension(B1);
    B3 = B1;
    ++B3.length;
    B3.breadth++;
    dimension(B3);
    return 0;
}
```

4) What is the output of the following program?

```
#include <iostream>
using namespace std;
int main(){
    int num[5];
    int* p;
    p = num;
    *p = 10;
    p++;
    *p = 20;
    p = &num[2];
    *p = 30;
    p = num + 3;
    *p = 40;
    p = num;
    *(p + 4) = 50;
    for (int i = 0; i < 5; i++)
        cout << num[i] << ", ";
    return 0;
}
```

5) What is the output of the following program?

```
#include <iostream>
using namespace std;
int main(){
    char a[] = { 'A', 'B', 'C', 'D' };
    char* ppp = &a[0];
    *ppp++;
    cout<<*++ppp << --*ppp;
    return 0;
}
```

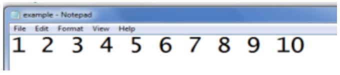
6) What is the output of the following program?

```
#include <iostream>
using namespace std;
int main(){
    int i = 6, *j, k;
    j = &i;
    cout << i * *j * i + *j;
    return 0;
}
```

2. Tracing a complete Program:

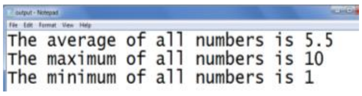
[*Student should trace this program, and write down the steps of running this program*]

- **Problem 1**: Write a program that reads numbers from a file, calculate the average, max and min of these numbers and write them in another file.
    - Reads numbers from a file example.txt:



```
1 2 3 4 5 6 7 8 9 10
```

    - Save avg, max and min in in another file:

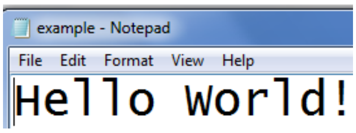


```
The average of all numbers is 5.5
The maximum of all numbers is 10
The minimum of all numbers is 1
```

Solution:

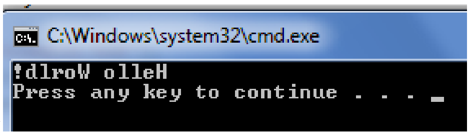
```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main () {
        int counter=0;
        double number, sum=0, average, max=0, min=100;
        ifstream read;
        ofstream write;
        read.open("example.txt");
        if (read.fail()){
                cout<< "Input file opening failed.\n";
                exit(0);
        }
        write.open("output.txt");
        if(write.fail()){
                cout << "Output file opening failed.\n";
                exit(1);
        }
        while (read>>number){
                counter++;
                sum=sum+number;
                if(number>max)
                        max = number;
                if(number<min)
                        min=number;
        }
        average=sum/counter;
        write<<"The average of all numbers is "<<average<<endl;
        write<<"The maximum of all numbers is "<<max<<endl;
        write<<"The minimum of all numbers is "<<min<<endl;
        read.close();
        write.close();
}
```

- **Problem 2**: Write a program that print the file content in reversed order. Do so by creating two functions; the first one called int characters_in_file() to count the characters in the file and the second one char character_at_position(int number) to print the characters in the reverse order.



```
Hello World!
```

➔

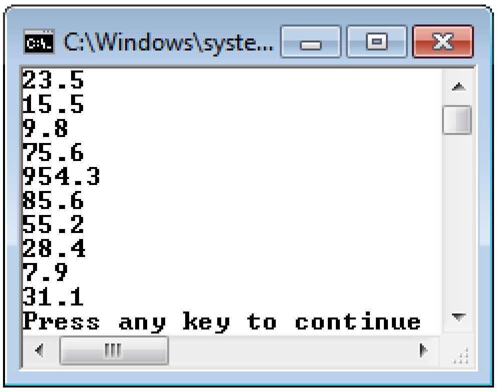


```
!dlroW olleH
Press any key to continue . . .
```

Solution:

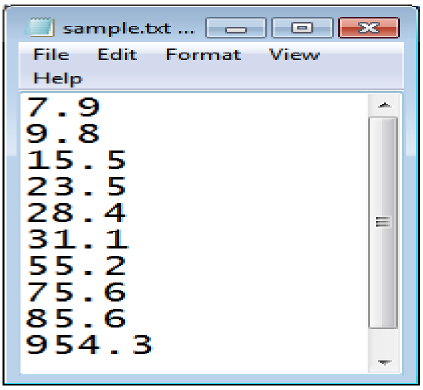
```
#include <iostream>
#include <fstream>
using namespace std;
int characters_in_file();
char character_at_position(int number);
void main (){
        int count;
        count = characters_in_file();
        for (int number= count; number > 0; number--){
                cout << character_at_position(number);
        }
        cout<<endl;
}
int characters_in_file(){
        char ch;
        ifstream in_stream;
        int count = 0;
        in_stream.open("example.txt");
        in_stream.get (ch);
        while (!in_stream.fail()) {
                count++;
                in_stream.get (ch);
        }
        in_stream.close();
        return count;
}
char character_at_position(int number){
        char ch;
        ifstream in_stream;
        in_stream.open("example.txt");
        for (int i=0; i<number; i++)
                in_stream.get(ch);
        in_stream.close();
        return ch;
}
```

- **Problem 3**: Write a C++ program that read from the user 10 double numbers and save these numbers as sorted in the file.



```
23.5
15.5
9.8
75.6
954.3
85.6
55.2
28.4
7.9
31.1
Press any key to continue
```

➔



```
7.9
9.8
15.5
23.5
28.4
31.1
55.2
75.6
85.6
954.3
```

Solution:

```
#include <iostream>
#include <fstream>
using namespace std;
void main() {
	const int num = 10;
	double arr[num];
	for (int i = 0; i < num; i++){
		cin >> arr[i];
	}
	int s = sizeof(arr) / sizeof(*arr);
	fstream fio;
	fio.open("sample.txt", ios::out);
	for (int i = 0; i < s - 1; i++){
		for (int j = i + 1; j < s; j++){
			if (arr[i] > arr[j]){
				double temp = arr[i];
				arr[i] = arr[j];
				arr[j] = temp;
			}
		}
	}
	for (int i = 0; i < s; i++){
		fio << arr[i] << endl;
	}
	fio.close();
}
```

- **Problem 4**: Write a C++ program to input elements in an array from user and sort all even and odd elements of the given array separately without using any other array. If minimum element of the array is even then all even elements should be placed in sorted order before odd elements otherwise all odd elements should be placed first. How to sort all even and odd elements of a given array separately in C programming.

  Input ➔`Input size of array: 10`

  ➔`Input elements of array: 0↲5↲1↲2↲3↲4↲6↲12↲10↲9`

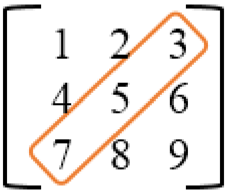
  Output:➔ `Output in sorted order: 0 2 4 6 10 12 1 3 5 9`

  Solution:

```
#include <iostream>
using namespace std;
const int MAX_SIZE = 1000; //Maximum size of the array
void arrange(int arr[], int len, int pivot);
void sort(int arr[], int start, int end);
void print(int arr[], int n);
void main(){
        int arr[MAX_SIZE], n;
        int pivot, evenCount, oddCount;
        pivot = 0;
        evenCount = oddCount = 0;
        cout<<"Enter size of the array: ";
        cin >> n;
        cout<< "Enter elements in the array: ";
        for (int i = 0; i<n; i++){
                // If current element is odd then increase pivot
                cin >> arr[i];
                if ((arr[i] % 2) == 1)                      oddCount++;
                else                                        evenCount++;
        }
        pivot = (evenCount > oddCount) ? evenCount : oddCount;
        print(arr, n);
        arrange(arr, n, pivot);
        cout << "\nElements after arranging even and odd elements separately\n";
        print(arr, n);
        sort(arr, pivot, n);
        sort(arr, 0, pivot);
        cout << "\nFinal array after sorting even and odd elements Separate\n";
        print(arr, n);
}
void arrange(int arr[], int len, int pivot){
        int temp;
        for (int i = 0; i<pivot; i++){
                if ((arr[i] % 2) == 1){
                        for (int j = pivot; j<len; j++){
                                if (!((arr[j] % 2) == 1)){
                                        temp = arr[j];
                                        arr[j] = arr[i];
                                        arr[i] = temp;
                                        break;
                                }
                        }
                }
        }
}
void sort(int arr[], int start, int end){
        int temp;
        for (int i = start; i < end; i++){
                for (int j = i + 1; j < end; j++){
                        if (arr[j] < arr[i]){
                                temp = arr[i];
                                arr[i] = arr[j];
                                arr[j] = temp;
                        }
                }
        }
}
void print(int arr[], int len){
        cout<<"Elements in the array: ";
        for (int i = 0; i<len; i++){
                cout<<arr[i]<<"    ";
        }
        cout << "\n";
}
```

- **Problem 5**: Write a C program to read elements in a matrix and find the sum of minor diagonal (opposite diagonal) elements. C program to calculate sum of minor diagonal elements. Logic to find sum of opposite diagonal elements of a matrix in C programming. Minor diagonal of a matrix A is a collection of elements Aij Such that i + j = N + 1 as follows:

```
Input elements in array: 1↲2↲3↲4↲5↲6↲7↲8↲9
Output: Sum of minor diagonal elements = 15
```



Solution:

```
#include <iostream>
using namespace std;
const int SIZE = 3; // Matrix size
void  main(){
        int A[SIZE][SIZE];
        int row, col, sum = 0;

        /* Input elements in matrix from user */
        cout<<"Enter elements in matrix of size : \n";
        for (int row = 0; row<SIZE; row++){
                for (int col = 0; col<SIZE; col++){
                        cin>>A[row][col];
                }
        }

        /* Find sum of minor diagonal elements */
        for (int row = 0; row<SIZE; row++){
                for (int col = 0; col<SIZE; col++){
                        /*
                        * If it is minor diagonal of matrix
                        * Minor diagonal: i+j == N + 1
                        * Since array elements starts from 0 hence i+j == (N + 1)-
2
                        */
                        if (row + col == ((SIZE + 1) - 2)){
                                sum += A[row][col];
                        }
                }
        }
        cout<<"\n Sum of minor diagonal elements = "<< sum;
}
```

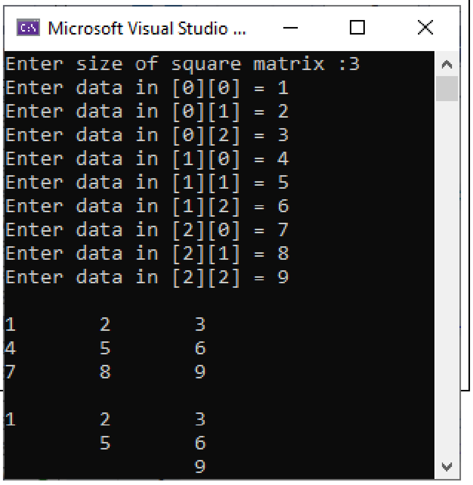
- **Problem 6**: Write a user defined function named `upper_half()` which takes a square two dimensional array, with size as argument and prints the upper half of the array.

e.g.,

```
2 3 1 5 0                                    2 3 1 5 0
7 1 5 3 1                                      1 5 3 1
2 5 7 8 1    →The output will be→                1 7 8
0 1 5 0 1                                          0 1
3 4 9 1 5                                            5
```

Solution:

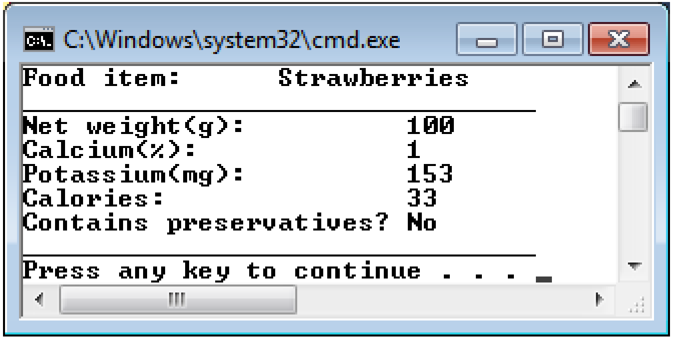
```cpp
#include <iostream>
using namespace std;
void input(int matrix[][10], int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            cout << "Enter data in [" << i << "][" << j << "] = ";
            cin >> matrix[i][j];
        }
    }
}
void display(int matrix[][10], int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            cout << matrix[i][j] << "\t";
        }
        cout << endl;
    }
}
void upper_half(int matrix[][10], int n) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (i <= j) cout << matrix[i][j] << "\t";
            else        cout << "\t";
        }
        cout << endl;
    }
}int main(){
    int s;
    int array[10][10];
    cout<<"Enter size of square matrix :";
    cin>>s;

    input(array, s);
    cout << endl;    display(array, s);
    cout << endl;    upper_half(array, s);
    return 0;
}
```



```
Enter size of square matrix :3
Enter data in [0][0] = 1
Enter data in [0][1] = 2
Enter data in [0][2] = 3
Enter data in [1][0] = 4
Enter data in [1][1] = 5
Enter data in [1][2] = 6
Enter data in [2][0] = 7
Enter data in [2][1] = 8
Enter data in [2][2] = 9

1       2       3
4       5       6
7       8       9

1       2       3
        5       6
                9
```

- **Problem 7**: Packaged food items have nutritional value data attached to their containers. You are to implement a struct to hold a food item's name and nutritional value data. The nutrients needed to be maintained (with corresponding suggested datatype) are net weight (float), potassium (int), calcium (int), contains preservatives (bool) and calories per serving (int). Write a function to display nutritional data of a given food product.

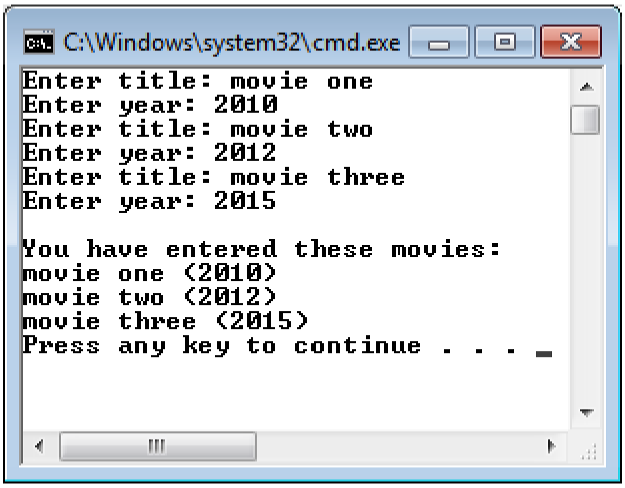
  Solution: [***Student should trace this program, and write down the steps of running this program***]



```
Food item:      Strawberries
________________________________
Net weight(g):          100
Calcium(%):             1
Potassium(mg):          153
Calories:               33
Contains preservatives? No
________________________________
Press any key to continue . . . _
```

```
#include <iostream>
#include <string>
using namespace std;
struct NutritionalValue{
       string foodName;
       float netWeight;
       int calcium;
       int potassium;
       int calories;
       bool hasPreservatives;
};
void displayNutritionalValue(NutritionalValue n){
       cout << "Food item:\t" << n.foodName << endl;
       cout << "________________________________" << endl;
       cout << "Net weight(g):\t\t" << n.netWeight << endl;
       cout << "Calcium(%):\t\t" << n.calcium << endl;
       cout << "Potassium(mg):\t\t" << n.potassium << endl;
       cout << "Calories:\t\t" << n.calories << endl;
       cout << "Contains preservatives?\t";
       if (n.hasPreservatives == true)
              cout << "Yes" << endl;
       else
              cout << "No" << endl;
       cout << "________________________________" << endl;
}
int main(){
       NutritionalValue strawberryNutrition;
       strawberryNutrition.foodName = "Strawberries";
       strawberryNutrition.netWeight = 100.0;
       strawberryNutrition.calcium = 1;
       strawberryNutrition.potassium = 153;
       strawberryNutrition.calories = 33;
       strawberryNutrition.hasPreservatives = false;
       displayNutritionalValue(strawberryNutrition);
       return 0;
}
```

- **Problem 8**: Write a C++ program to read the information of three movies, each movie has a title and year of production. Each movie is a structure, and the three movie should be saved as an array of structures. After reading the array, print each movie by a function printmovie() by a loop in the main program:



```
Enter title: movie one
Enter year: 2010
Enter title: movie two
Enter year: 2012
Enter title: movie three
Enter year: 2015

You have entered these movies:
movie one (2010)
movie two (2012)
movie three (2015)
Press any key to continue . . .
```

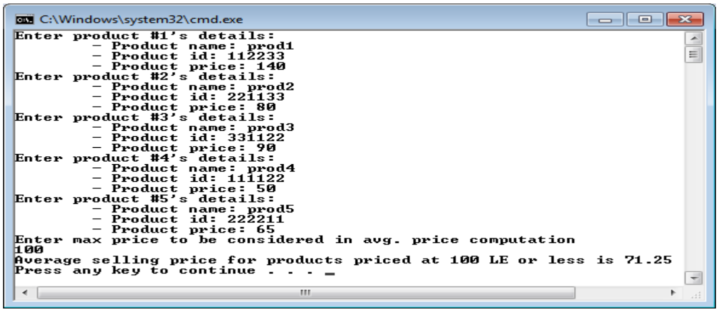
Solution:

```cpp
// array of structures
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
struct movies_t {
    string title;
    int year;
};
void printmovie(movies_t movie);
int main(){
    movies_t films[3];
    string myStr = "";
    int n;
    for (n = 0; n<3; n++){
            cout << "Enter title: ";
            getline(cin, films[n].title);
            cout << "Enter year: ";
            getline(cin, myStr);
            films[n].year = stoi(myStr);
            //stringstream(mystr) >> ;
    }
    cout << "\nYou have entered these movies:\n";
    for (n = 0; n<3; n++)
            printmovie(films[n]);
    return 0;
}
void printmovie(movies_t movie){
    cout << movie.title;
    cout << " (" << movie.year << ")\n";
}
```

- **Problem 9**: A shop sells 5 different products. Represent each product as a struct, and the product object should hold information regarding product name, id and price. Use appropriate data types for each data field as needed. Get from the user the 5 products and an integer value of name maxPrice. Then write a function to compute and return the average of all products' price, the price of these products must be lower than a certain price (maxPrice):

  Solution:

```
#include <iostream>
#include <string>
using namespace std;
struct ProductStruct{
    string name;
    int id;
    float price;
};
```

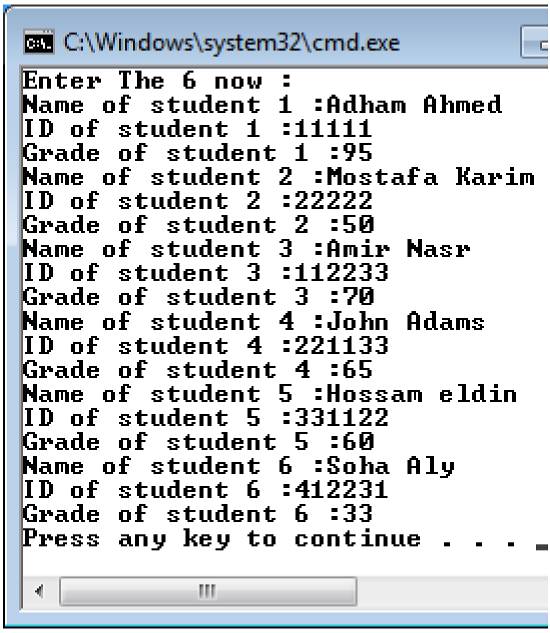


```
Enter product #1's details:
        - Product name: prod1
        - Product id: 112233
        - Product price: 140
Enter product #2's details:
        - Product name: prod2
        - Product id: 221133
        - Product price: 80
Enter product #3's details:
        - Product name: prod3
        - Product id: 331122
        - Product price: 90
Enter product #4's details:
        - Product name: prod4
        - Product id: 111122
        - Product price: 50
Enter product #5's details:
        - Product name: prod5
        - Product id: 222211
        - Product price: 65
Enter max price to be considered in avg. price computation
100
Average selling price for products priced at 100 LE or less is 71.25
Press any key to continue . . .
```

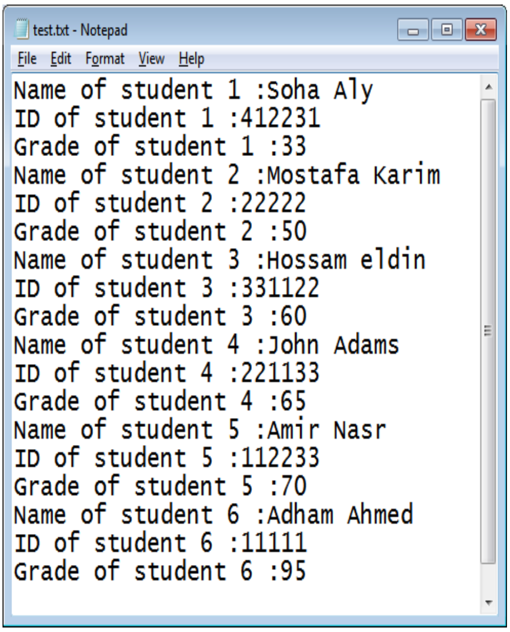
```
/* This function takes an array of 5 products then computes, and returns, the
average of all products priced under a certain price (maxPrice)*/
float returnAvg(ProductStruct products[5], int maxPrice){
    float sum = 0, count = 0;
    for (int i = 0; i < 5; i++){
           // check only the prices less than or equal the specified max price
entered by the user
           if (products[i].price <= maxPrice){
                  sum += products[i].price;
                  count++;
                  // Counts the number of the Prices less than or equal the
specified max price to be used for the avg.
           }
    }
    return sum / count;
}

void main(){
    ProductStruct products[5]; // Products Array of Type "ProductStruct"
    int maxPrice;
    // accept user input
    for (int i = 0; i < 5; i++){
           cout << "Enter product #" << i + 1 << "'s details:" << endl;
           cout << "\t- Product name: ";
           cin >> products[i].name;
           cout << "\t- Product id: ";
           cin >> products[i].id;
           cout << "\t- Product price: ";
           cin >> products[i].price;
    }
    // accept user choice of max price
    cout << "Enter max price to be considered in avg. price computation" << endl;
    cin >> maxPrice;
    // compute and display average selling price
    cout << "Average selling price for products priced at " << maxPrice << " LE or
less is " << returnAvg(products, maxPrice) << endl;
}
```

- **Problem 10**: Define structure named as student that contains string "studentName" and int "studentId" and float "studentGrade".
  - Define a global constant "numOfStudent" of value 6.
  - Define a function "readStudents" that accepts the array of students as a paramter, and reads the array from the user input in the cmd.
  - Define a function "sortStudents" that accepts the array of students as a paramter, and sort the array according to students grades in an ascending order.
  - Define a function "saveStudents" that accepts the array of students as a paramter, and save the array in a file named as "test.txt".
  - In the main function, define an array of students "students", the size of the array is "numOfStudent"; In the main function, Call the functions "readStudents", "sortStudents", "saveStudents" in this order. Send to these functions the array "students" as a parameter.

  - The input of the program and the output test.txt file is as follows:



```
Enter The 6 now :
Name of student 1 :Adham Ahmed
ID of student 1 :11111
Grade of student 1 :95
Name of student 2 :Mostafa Karim
ID of student 2 :22222
Grade of student 2 :50
Name of student 3 :Amir Nasr
ID of student 3 :112233
Grade of student 3 :70
Name of student 4 :John Adams
ID of student 4 :221133
Grade of student 4 :65
Name of student 5 :Hossam eldin
ID of student 5 :331122
Grade of student 5 :60
Name of student 6 :Soha Aly
ID of student 6 :412231
Grade of student 6 :33
Press any key to continue . . .
```

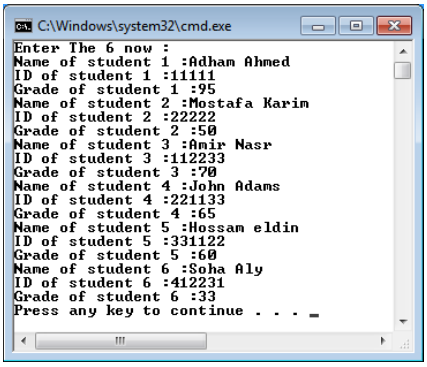


```
Name of student 1 :Soha Aly
ID of student 1 :412231
Grade of student 1 :33
Name of student 2 :Mostafa Karim
ID of student 2 :22222
Grade of student 2 :50
Name of student 3 :Hossam eldin
ID of student 3 :331122
Grade of student 3 :60
Name of student 4 :John Adams
ID of student 4 :221133
Grade of student 4 :65
Name of student 5 :Amir Nasr
ID of student 5 :112233
Grade of student 5 :70
Name of student 6 :Adham Ahmed
ID of student 6 :11111
Grade of student 6 :95
```

Solution:

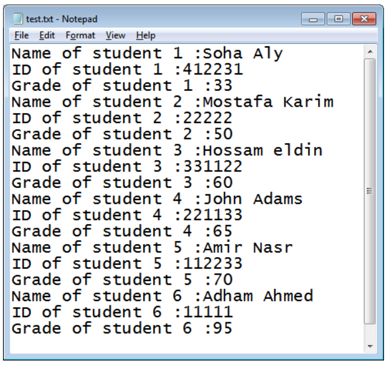
```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
const int numOfStudent = 6;
struct student{
        string studentName;
        int studentId;
        float studentGrade;
};
void readStudents(student students[]){
        cout << "Enter The " << numOfStudent << " now : \n";
        for (int i = 0; i < numOfStudent; i++){
                cout << "Name of student " << (i + 1) << " :";
                getline(cin, students[i].studentName);
                cout << "ID of student " << (i + 1) << " :";
                cin >> students[i].studentId;
                cout << "Grade of student " << (i + 1) << " :";
                cin >> students[i].studentGrade;
                cin.ignore();
                }
}
void sortStudents(student students[]){
        for (int i = 0; i < numOfStudent - 1; i++){
                for (int j = i + 1; j < numOfStudent; j++){
                        if (students[i].studentGrade > students[j].studentGrade){
                                student temp = students[i];
                                students[i] = students[j];
                                students[j] = temp;
                        }
                }
        }
}
void saveStudents(student students[]) {
        fstream fio;
        fio.open("test.txt", ios::out);
        for (int i = 0; i < numOfStudent; i++){
                fio << "Name of student " << (i + 1) << " :";
                fio << students[i].studentName << endl;
                fio << "ID of student " << (i + 1) << " :";
                fio << students[i].studentId << endl;
                fio << "Grade of student " << (i + 1) << " :";
                fio << students[i].studentGrade << endl;
        }
        fio.close();
}
void main(){
        student students[numOfStudent];
        readStudents(students);
        sortStudents(students);
        saveStudents(students);
}
```

- **Problem 11:** Write a C++ program that gets from the user 6 students, each student has a name, id and grade. Then sort these students according to their grades in an ascending order. Then save these students in a file named as test.txt. The figure below in the left shows the input from the user to command prompt, while the figure on the right shows the saved array in the test.txt file.



```
Enter The 6 now :
Name of student 1 :Adham Ahmed
ID of student 1 :11111
Grade of student 1 :95
Name of student 2 :Mostafa Karim
ID of student 2 :22222
Grade of student 2 :50
Name of student 3 :Amir Nasr
ID of student 3 :112233
Grade of student 3 :70
Name of student 4 :John Adams
ID of student 4 :221133
Grade of student 4 :65
Name of student 5 :Hossam eldin
ID of student 5 :331122
Grade of student 5 :60
Name of student 6 :Soha Aly
ID of student 6 :412231
Grade of student 6 :33
Press any key to continue . . .
```

➔



```
Name of student 1 :Soha Aly
ID of student 1 :412231
Grade of student 1 :33
Name of student 2 :Mostafa Karim
ID of student 2 :22222
Grade of student 2 :50
Name of student 3 :Hossam eldin
ID of student 3 :331122
Grade of student 3 :60
Name of student 4 :John Adams
ID of student 4 :221133
Grade of student 4 :65
Name of student 5 :Amir Nasr
ID of student 5 :112233
Grade of student 5 :70
Name of student 6 :Adham Ahmed
ID of student 6 :11111
Grade of student 6 :95
```

The detailed steps that you should implement in your program are as follows:

1. Define structure named as student that contains string "studentName" and int "studentId" and float "studentGrade". And define a global constant "numOfStudent" of value 6.

2. Define a function "readStudents" that accepts the array of students as a parameter, and reads the array from the user input in the cmd.

3. Define a function "sortStudents" that accepts the array of students as a parameter, and sort the array according to student’s grades in an ascending order.

4. Define a function "saveStudents" that accepts the array of students as a parameter, and save the array in a file named as "test.txt".

5. In the main function, define an array of students "students", the size of the array is "numOfStudent". In the main function, Call the functions "readStudents", "sortStudents", "saveStudents" in this order. Send to these functions the array "students" as a parameter.

Solution:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
const int numOfStudent = 6;
struct student{
       string studentName;
       int studentId;
       float studentGrade;
};
void readStudents(student students[]){
       cout << "Enter The " << numOfStudent << " now : \n";
       for (int i = 0; i < numOfStudent; i++){
              cout << "Name of student " << (i + 1) << " :";
              getline(cin, students[i].studentName);
              cout << "ID of student " << (i + 1) << " :";
              cin >> students[i].studentId;
              cout << "Grade of student " << (i + 1) << " :";
              cin >> students[i].studentGrade;
              cin.ignore();
       }
}
void sortStudents(student students[]){
       for (int i = 0; i < numOfStudent - 1; i++){
         for (int j = i + 1; j < numOfStudent; j++){
           if (students[i].studentGrade > students[j].studentGrade){
                     student temp = students[i];
                     students[i] = students[j];
                     students[j] = temp;
           }
         }
       }
}
void saveStudents(student students[]) {
       fstream fio;
       fio.open("test.txt", ios::out);
       for (int i = 0; i < numOfStudent; i++){
              fio << "Name of student " << (i + 1) << " :";
              fio << students[i].studentName << endl;
              fio << "ID of student " << (i + 1) << " :";
              fio << students[i].studentId << endl;
              fio << "Grade of student " << (i + 1) << " :";
              fio << students[i].studentGrade << endl;
       }
       fio.close();
}
void main(){
       student students[numOfStudent];
       readStudents(students);
       sortStudents(students);
       saveStudents(students);
}
```

- **Problem 12:** Write a C++ function to sort an array of ten integer values in ascending order using pointers. The function will accept two arguments-- a pointer that points to the array and the array size. The function returns a pointer that points to the sorted array.

  Solution:

```
#include<iostream>
using namespace std;
int *sortAsc(int *p, int size);
int main()
{
    int arr[] = { 23, 34, 2, 3, 5, 12, 42, 56, 89, 8 };
    int *p = sortAsc(arr, 10);
    //output the sorted array
    int i;
    for (i = 0; i<10; i++)
        cout << *(p + i) << endl;
    return 0;

}
int *sortAsc(int *p, int n){
    int i, j;
    for (i = 0; i<n; i++)
    for (j = i + 1; j<n; j++)
    if (*(p + j)<*(p + i))
    {
        int temp = *(p + j);
        *(p + j) = *(p + i);
        *(p + i) = temp;
    }
    return p;
}
```