Programming Fundamentals

Bazla Tariq

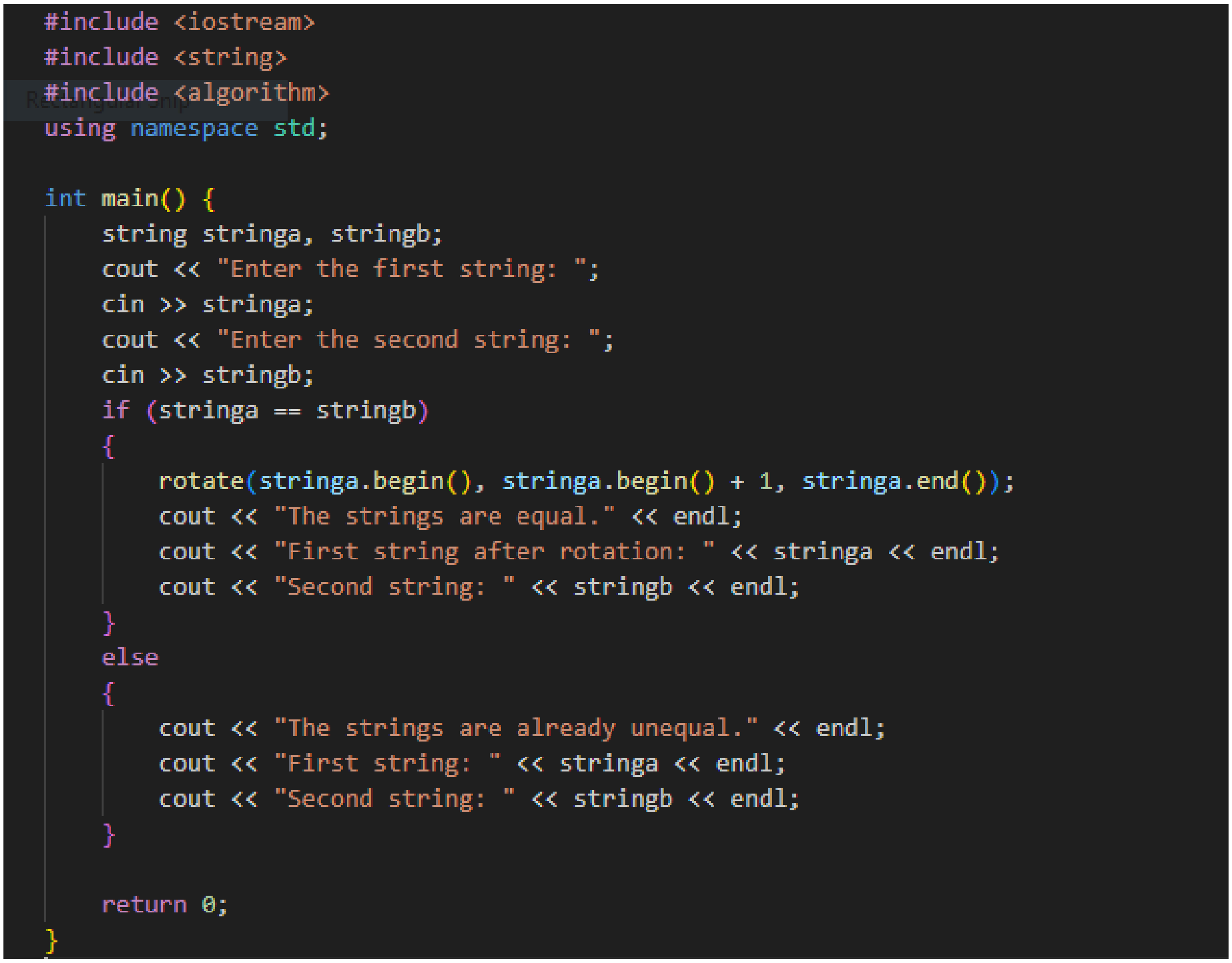
School of Mechanical and Manufacturing Engineering, NUST

Fundamentals of Programming

Lab Engineer Laiba Malik

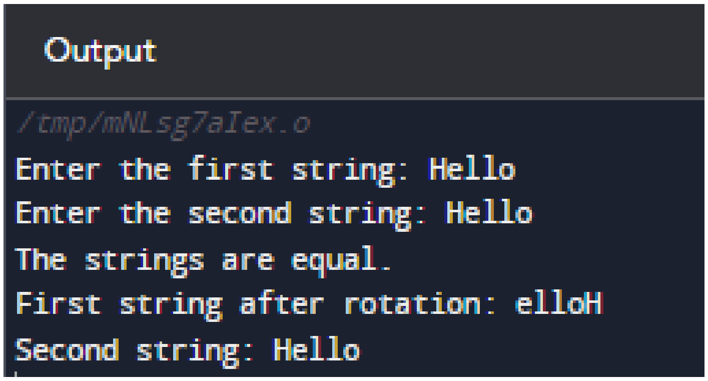
December 20, 2023

1. Write a C++ program, take two strings as input from user and check if both strings are equal or not. If they are equal make them unequal by rotating string. e.g., Hello is turned into olleH etc.



```cpp
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

int main() {
    string stringa, stringb;
    cout << "Enter the first string: ";
    cin >> stringa;
    cout << "Enter the second string: ";
    cin >> stringb;
    if (stringa == stringb)
    {
        rotate(stringa.begin(), stringa.begin() + 1, stringa.end());
        cout << "The strings are equal." << endl;
        cout << "First string after rotation: " << stringa << endl;
        cout << "Second string: " << stringb << endl;
    }
    else
    {
        cout << "The strings are already unequal." << endl;
        cout << "First string: " << stringa << endl;
        cout << "Second string: " << stringb << endl;
    }

    return 0;
}
```

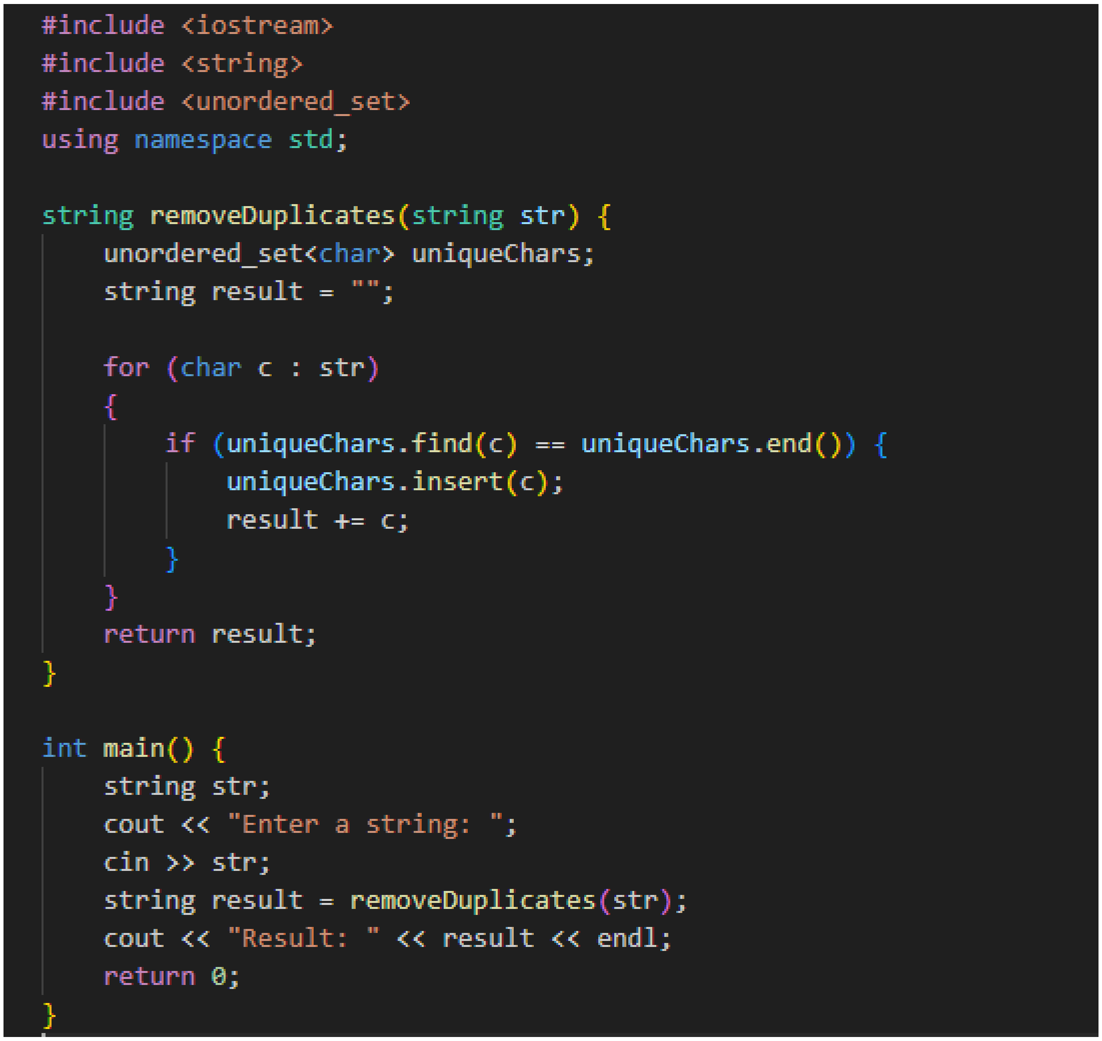
Figure 1: Question no: 1



```
/tmp/mNLsg7aIex.o
Enter the first string: Hello
Enter the second string: Hello
The strings are equal.
First string after rotation: elloH
Second string: Hello
```

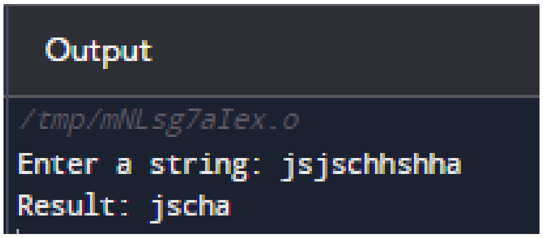
Figure 2: Output of Question 1

2. Write a C++program for a string which may contain lowercase and uppercase characters. The task is to remove all duplicate characters from the string and find the resultant string.



```cpp
#include <iostream>
#include <string>
#include <unordered_set>
using namespace std;

string removeDuplicates(string str) {
    unordered_set<char> uniqueChars;
    string result = "";

    for (char c : str)
    {
        if (uniqueChars.find(c) == uniqueChars.end()) {
            uniqueChars.insert(c);
            result += c;
        }
    }
    return result;
}

int main() {
    string str;
    cout << "Enter a string: ";
    cin >> str;
    string result = removeDuplicates(str);
    cout << "Result: " << result << endl;
    return 0;
}
```

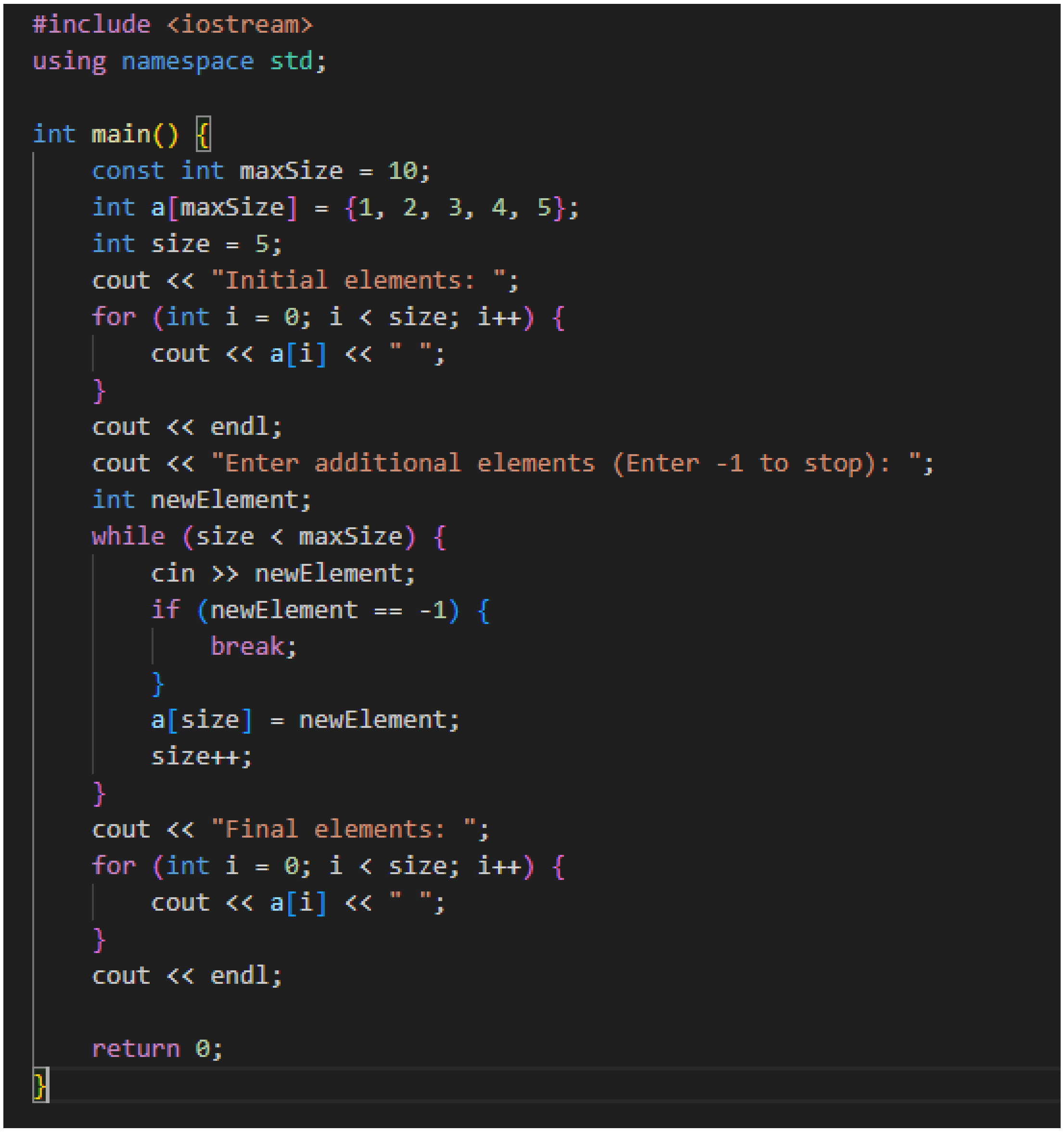
Figure 3: Question no: 2



```
/tmp/mNLsg7aIex.o
Enter a string: jsjschhshha
Result: jscha
```

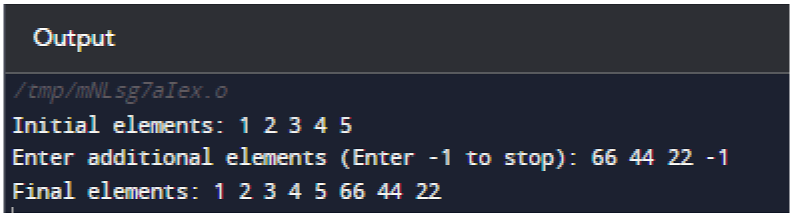
Figure 4: Answer Question no: 2

3. Suppose an integer array a[5] = {1,2,3,4,5}. Add more elements to it and display them in C++.



```cpp
#include <iostream>
using namespace std;

int main() {
    const int maxSize = 10;
    int a[maxSize] = {1, 2, 3, 4, 5};
    int size = 5;
    cout << "Initial elements: ";
    for (int i = 0; i < size; i++) {
        cout << a[i] << " ";
    }
    cout << endl;
    cout << "Enter additional elements (Enter -1 to stop): ";
    int newElement;
    while (size < maxSize) {
        cin >> newElement;
        if (newElement == -1) {
            break;
        }
        a[size] = newElement;
        size++;
    }
    cout << "Final elements: ";
    for (int i = 0; i < size; i++) {
        cout << a[i] << " ";
    }
    cout << endl;

    return 0;
}
```

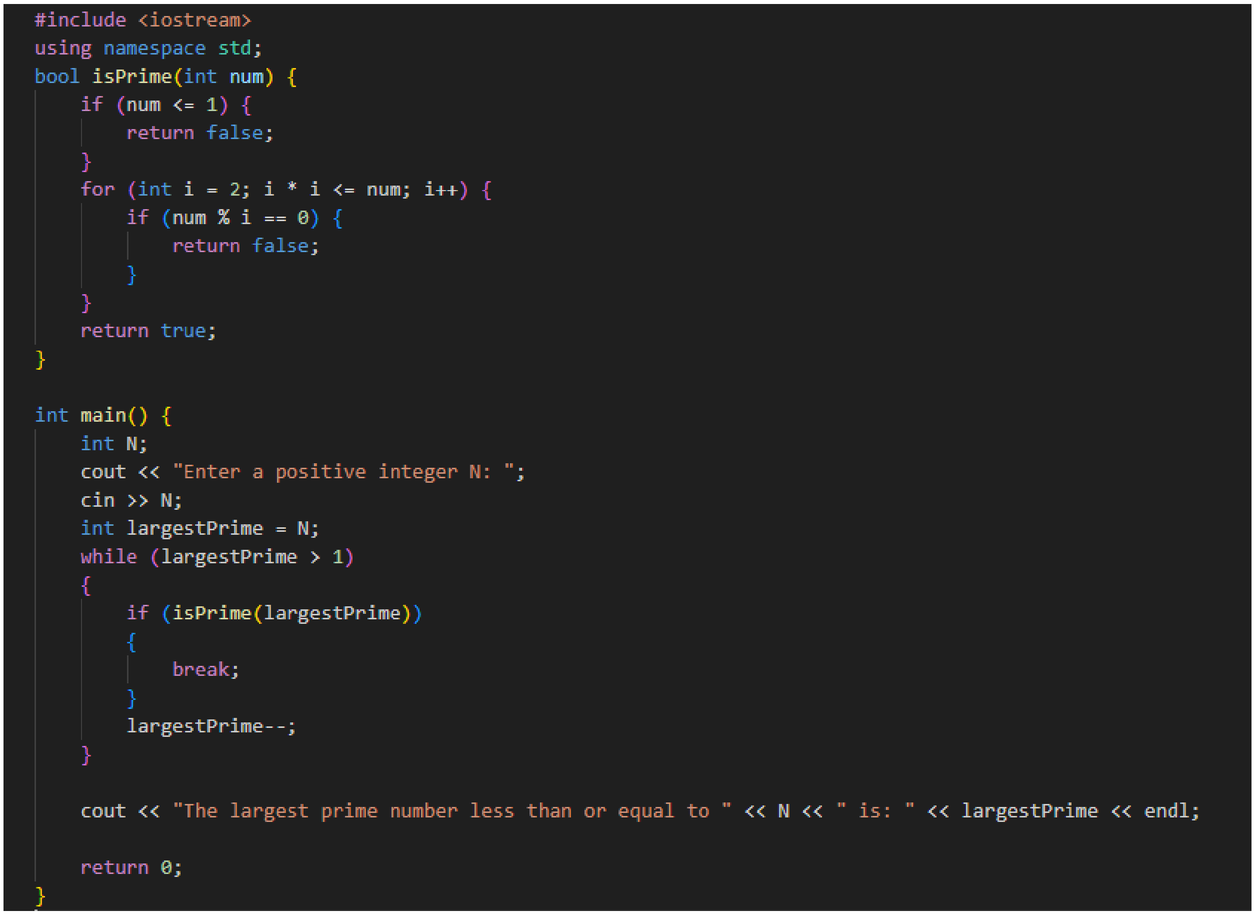
Figure 5: Question no: 3



```
/tmp/mNLsg7aIex.o
Initial elements: 1 2 3 4 5
Enter additional elements (Enter -1 to stop): 66 44 22 -1
Final elements: 1 2 3 4 5 66 44 22
```

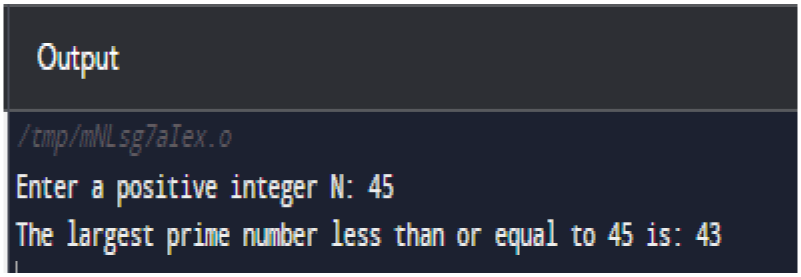
Figure 6: Output of Question no: 3

4. Write a C++ program that uses a while loop to find the largest prime number less than a given positive integer N. Your program should take the value of N as input from the user and then find the largest prime number less than or equal to N. You are not allowed to use any library or pre-existing functions to check for prime numbers.



```cpp
#include <iostream>
using namespace std;
bool isPrime(int num) {
    if (num <= 1) {
        return false;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

int main() {
    int N;
    cout << "Enter a positive integer N: ";
    cin >> N;
    int largestPrime = N;
    while (largestPrime > 1)
    {
        if (isPrime(largestPrime))
        {
            break;
        }
        largestPrime--;
    }

    cout << "The largest prime number less than or equal to " << N << " is: " << largestPrime << endl;

    return 0;
}
```

Figure 7: Question no: 4

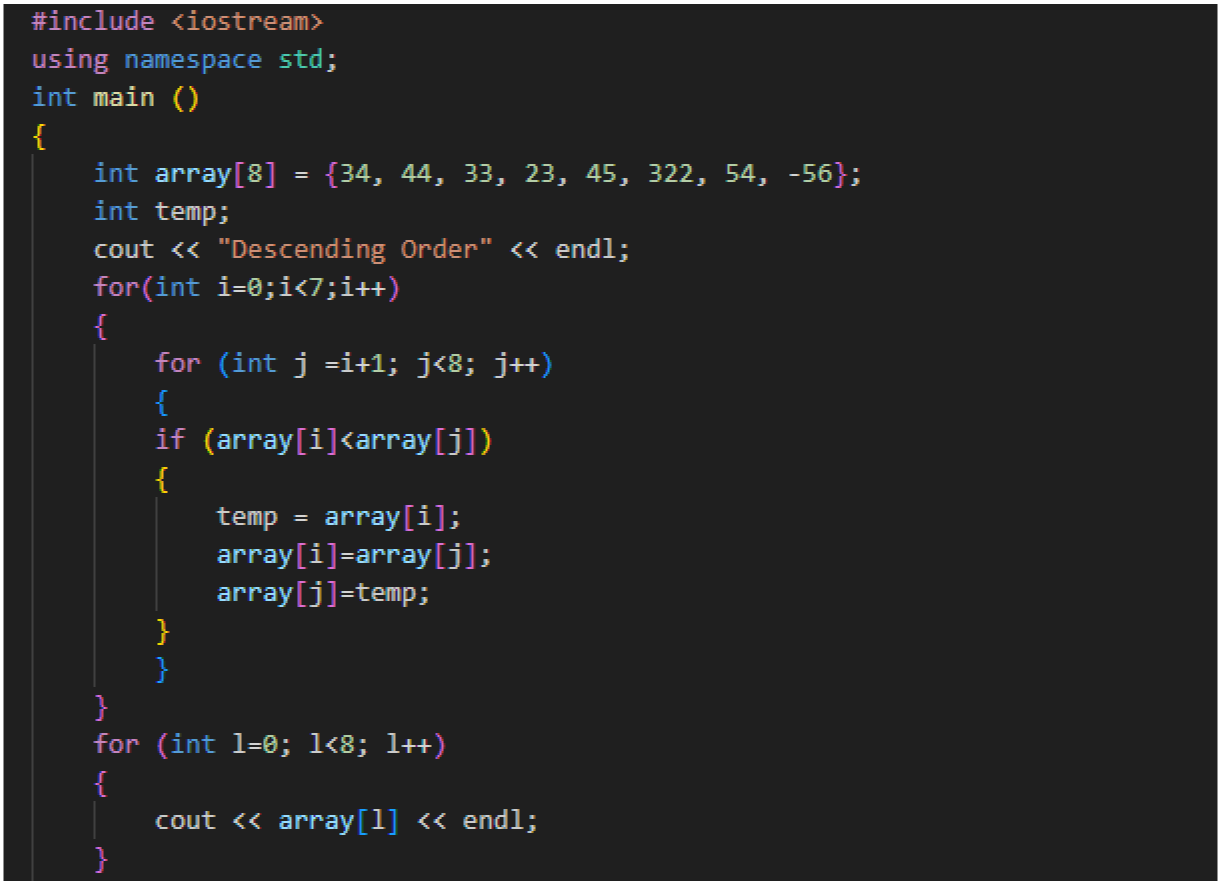


Output

```
/tmp/mNLsg7aIex.o
Enter a positive integer N: 45
The largest prime number less than or equal to 45 is: 43
```

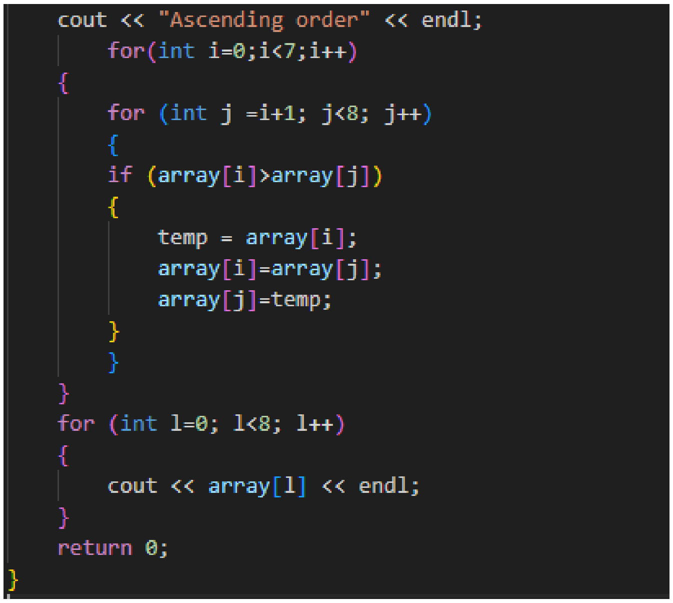
Figure 8: Output of Question no: 04

5. Implement Bubble Sort on an array of 6 integers.



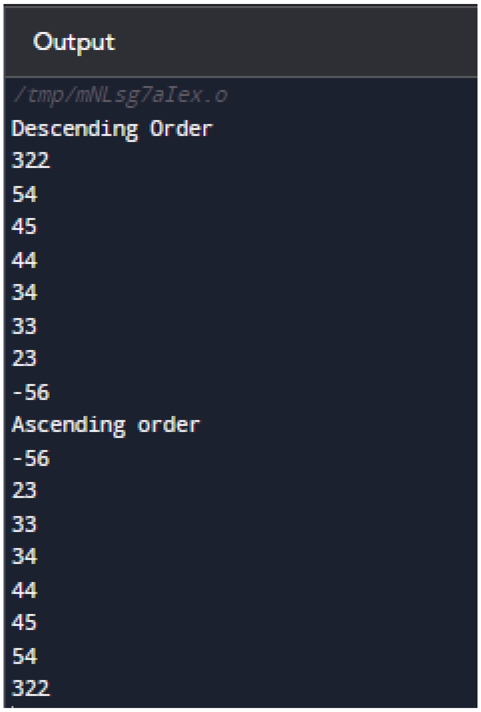
```
#include <iostream>
using namespace std;
int main ()
{
    int array[8] = {34, 44, 33, 23, 45, 322, 54, -56};
    int temp;
    cout << "Descending Order" << endl;
    for(int i=0;i<7;i++)
    {
        for (int j =i+1; j<8; j++)
        {
        if (array[i]<array[j])
        {
            temp = array[i];
            array[i]=array[j];
            array[j]=temp;
        }
        }
    }
    for (int l=0; l<8; l++)
    {
        cout << array[l] << endl;
    }
```

Figure 9: Question no: 5 (a)



```
    cout << "Ascending order" << endl;
        for(int i=0;i<7;i++)
    {
        for (int j =i+1; j<8; j++)
        {
        if (array[i]>array[j])
        {
            temp = array[i];
            array[i]=array[j];
            array[j]=temp;
        }
        }
    }
    for (int l=0; l<8; l++)
    {
        cout << array[l] << endl;
    }
    return 0;
}
```

Figure 10: Question no: 5 (b)

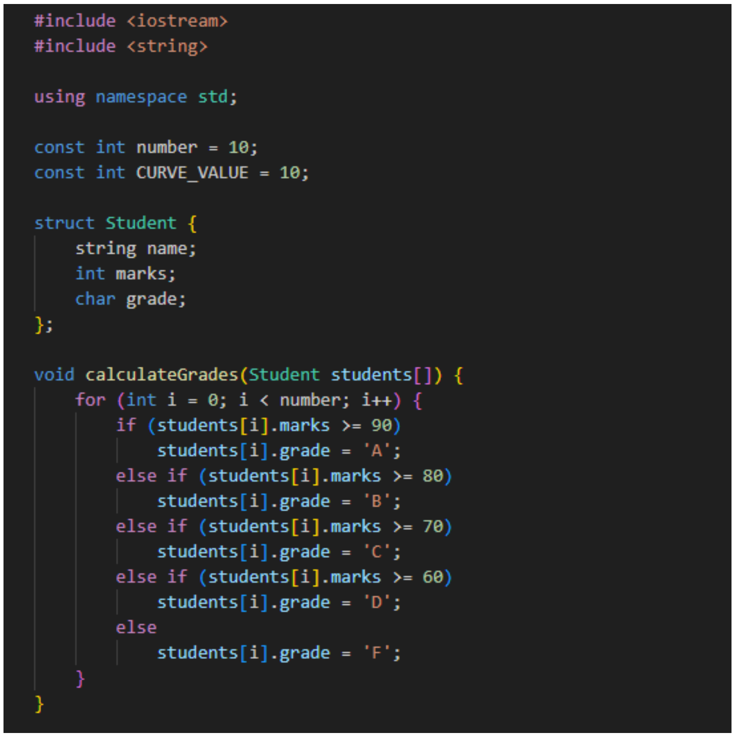


```
/tmp/mNLsg7aIex.o
Descending Order
322
54
45
44
34
33
23
-56
Ascending order
-56
23
33
34
44
45
54
322
```

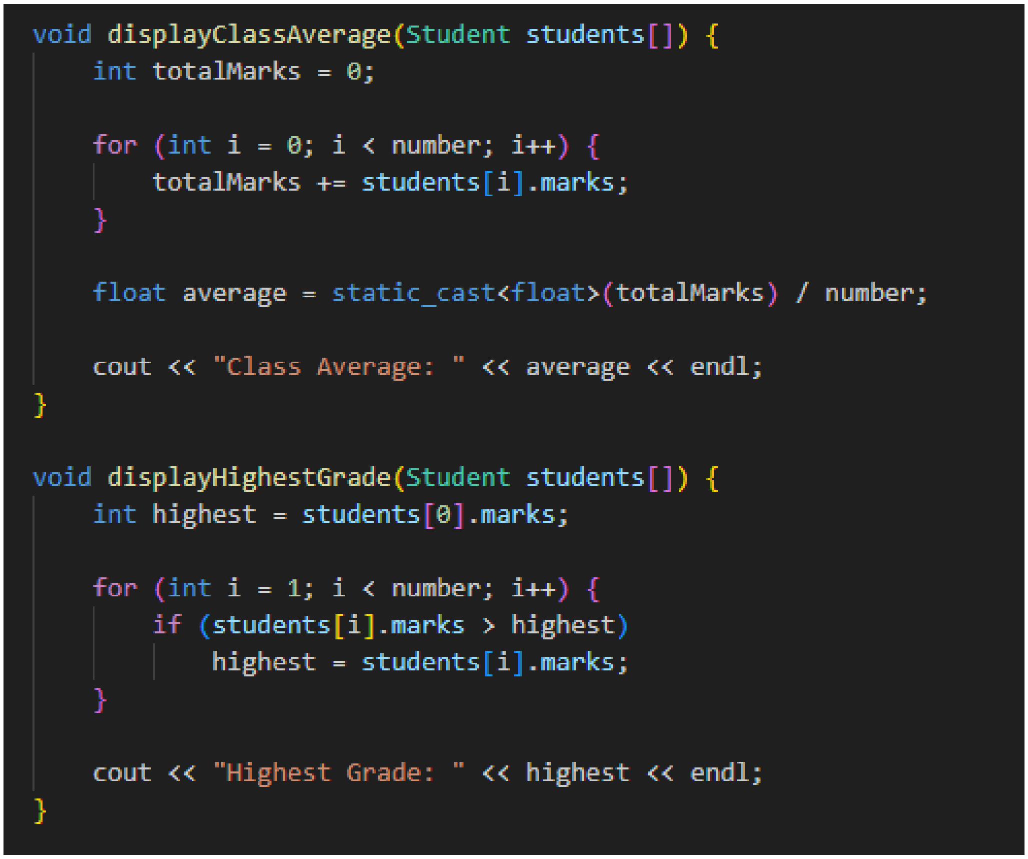
Figure 11: Output of Question no: 5

6. Solve any Aerospace/Real Life Problem using C++ Programming.

Problem: A program that stores the names and marks of students in an array and print grades first using the absolute, and then the relative grading system.

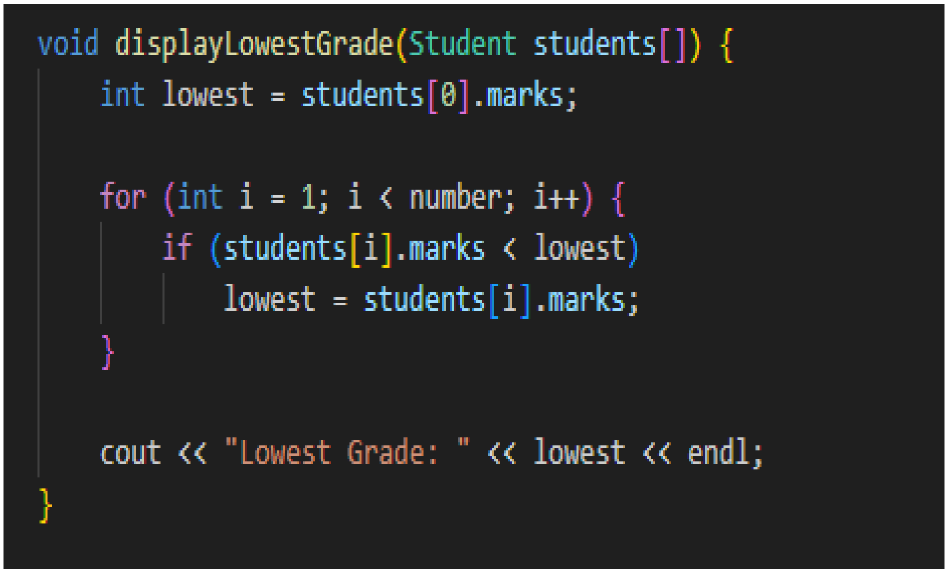


```cpp
#include <iostream>
#include <string>

using namespace std;

const int number = 10;
const int CURVE_VALUE = 10;

struct Student {
    string name;
    int marks;
    char grade;
};

void calculateGrades(Student students[]) {
    for (int i = 0; i < number; i++) {
        if (students[i].marks >= 90)
            students[i].grade = 'A';
        else if (students[i].marks >= 80)
            students[i].grade = 'B';
        else if (students[i].marks >= 70)
            students[i].grade = 'C';
        else if (students[i].marks >= 60)
            students[i].grade = 'D';
        else
            students[i].grade = 'F';
    }
}
```



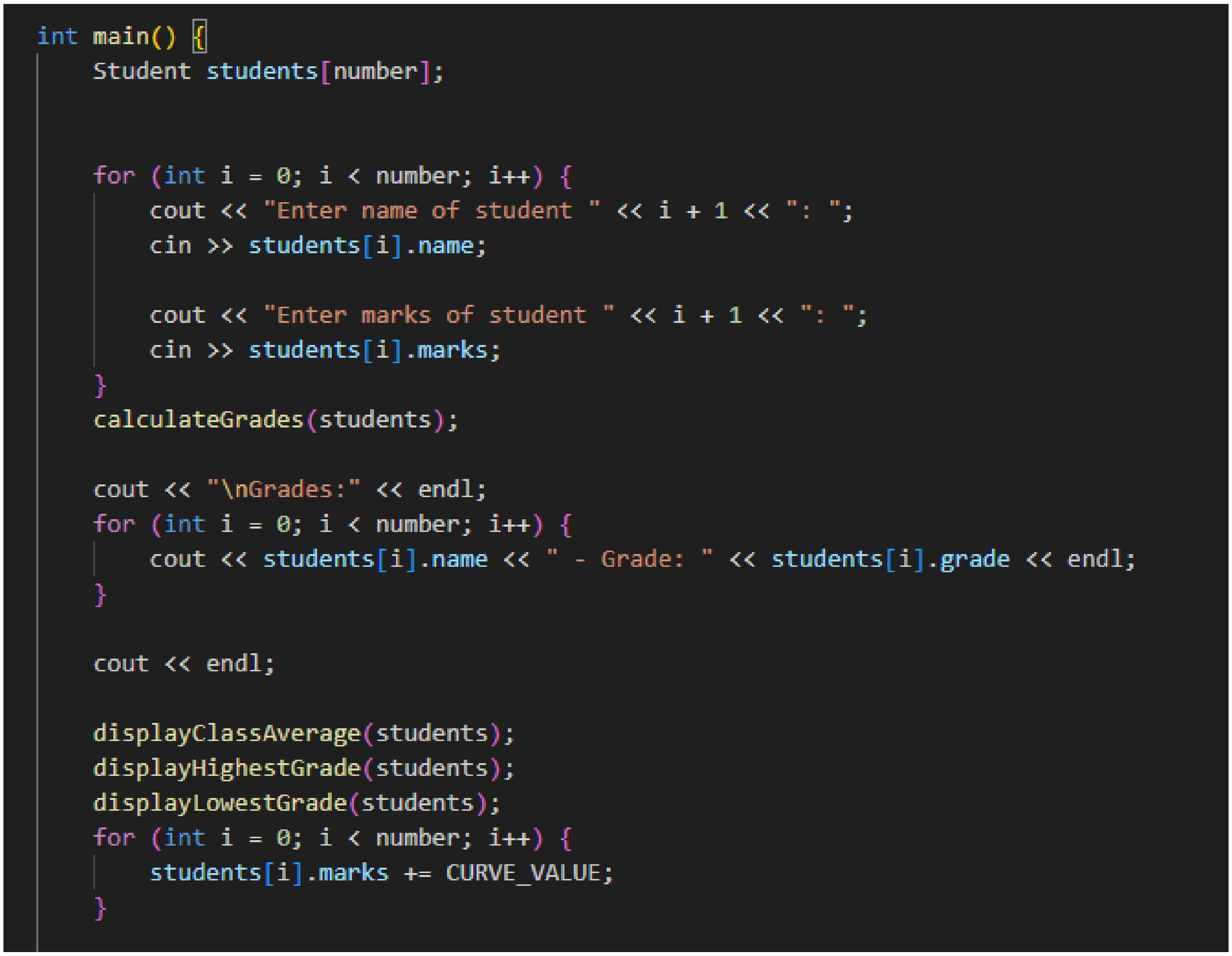
```cpp
void displayClassAverage(Student students[]) {
    int totalMarks = 0;

    for (int i = 0; i < number; i++) {
        totalMarks += students[i].marks;
    }

    float average = static_cast<float>(totalMarks) / number;

    cout << "Class Average: " << average << endl;
}

void displayHighestGrade(Student students[]) {
    int highest = students[0].marks;

    for (int i = 1; i < number; i++) {
        if (students[i].marks > highest)
            highest = students[i].marks;
    }

    cout << "Highest Grade: " << highest << endl;
}
```

Figure 12: Q 6



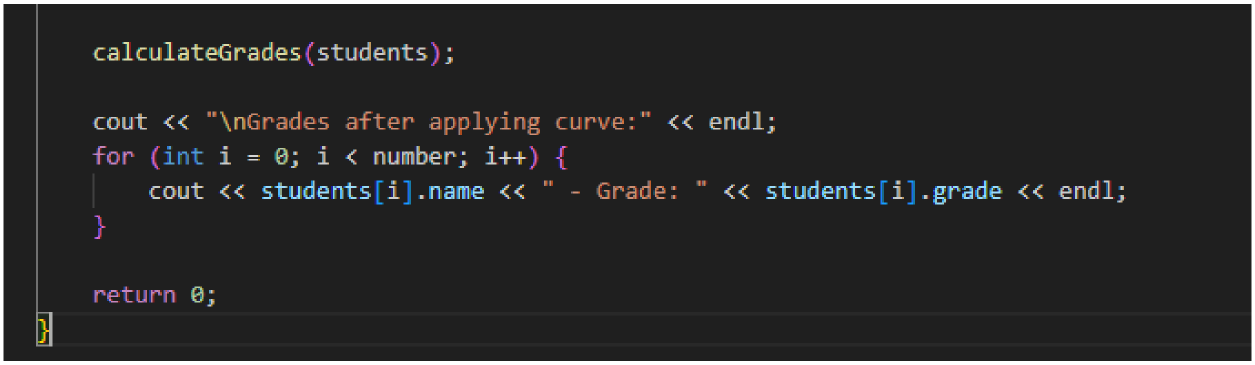
```cpp
void displayLowestGrade(Student students[]) {
    int lowest = students[0].marks;

    for (int i = 1; i < number; i++) {
        if (students[i].marks < lowest)
            lowest = students[i].marks;
    }

    cout << "Lowest Grade: " << lowest << endl;
}
```

Figure 13: Q 6



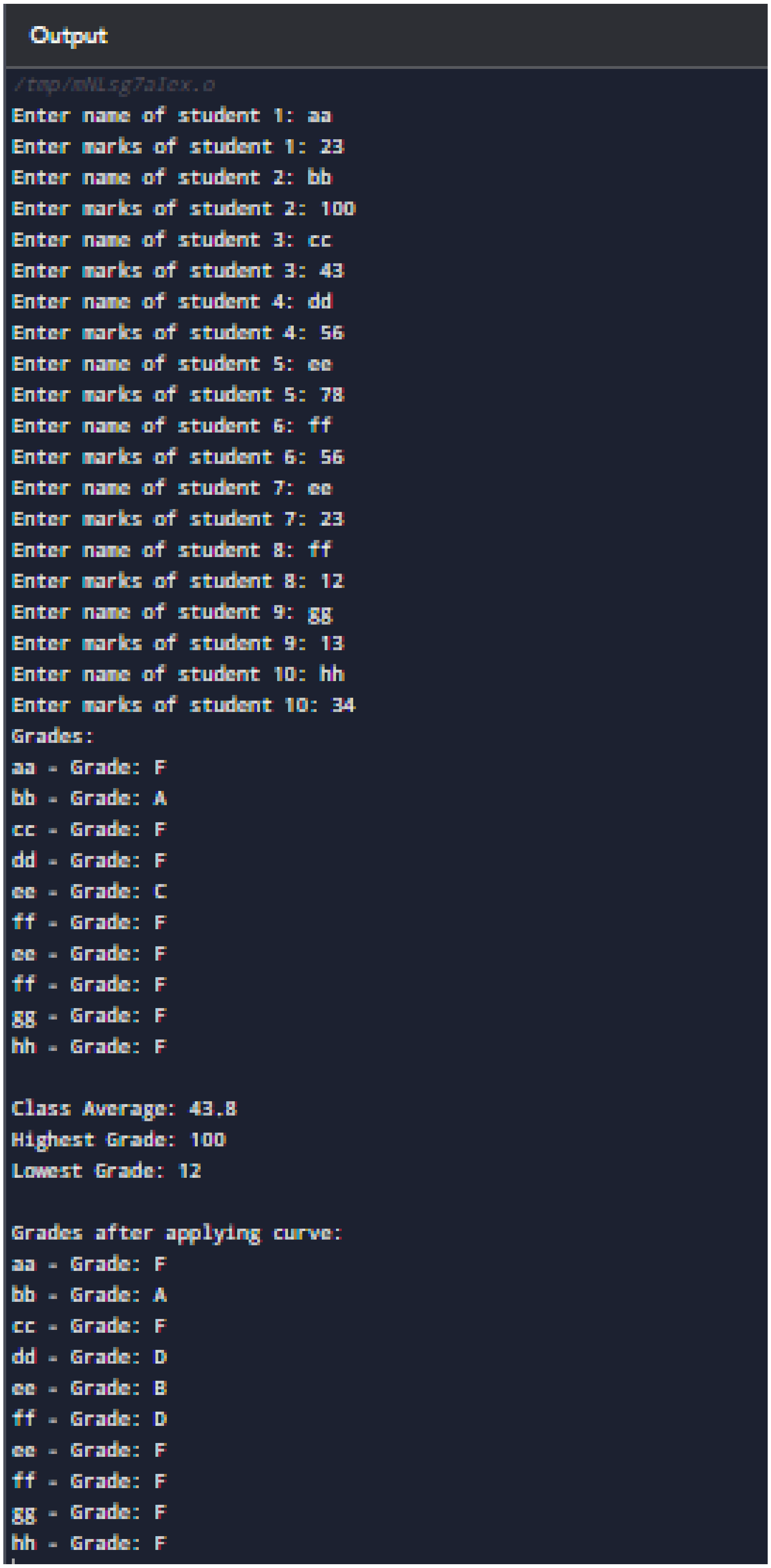
```cpp
int main() {
    Student students[number];


    for (int i = 0; i < number; i++) {
        cout << "Enter name of student " << i + 1 << ": ";
        cin >> students[i].name;

        cout << "Enter marks of student " << i + 1 << ": ";
        cin >> students[i].marks;
    }
    calculateGrades(students);

    cout << "\nGrades:" << endl;
    for (int i = 0; i < number; i++) {
        cout << students[i].name << " - Grade: " << students[i].grade << endl;
    }

    cout << endl;

    displayClassAverage(students);
    displayHighestGrade(students);
    displayLowestGrade(students);
    for (int i = 0; i < number; i++) {
        students[i].marks += CURVE_VALUE;
    }
```

Figure 14: Q 6



```cpp
    calculateGrades(students);

    cout << "\nGrades after applying curve:" << endl;
    for (int i = 0; i < number; i++) {
        cout << students[i].name << " - Grade: " << students[i].grade << endl;
    }

    return 0;
}
```

Figure 15: Q 6



```
/tmp/mNLsg7aIex.o
Enter name of student 1: aa
Enter marks of student 1: 23
Enter name of student 2: bb
Enter marks of student 2: 100
Enter name of student 3: cc
Enter marks of student 3: 43
Enter name of student 4: dd
Enter marks of student 4: 56
Enter name of student 5: ee
Enter marks of student 5: 78
Enter name of student 6: ff
Enter marks of student 6: 56
Enter name of student 7: ee
Enter marks of student 7: 23
Enter name of student 8: ff
Enter marks of student 8: 12
Enter name of student 9: gg
Enter marks of student 9: 13
Enter name of student 10: hh
Enter marks of student 10: 34
Grades:
aa - Grade: F
bb - Grade: A
cc - Grade: F
dd - Grade: F
ee - Grade: C
ff - Grade: F
ee - Grade: F
ff - Grade: F
gg - Grade: F
hh - Grade: F

Class Average: 43.8
Highest Grade: 100
Lowest Grade: 12

Grades after applying curve:
aa - Grade: F
bb - Grade: A
cc - Grade: F
dd - Grade: D
ee - Grade: B
ff - Grade: D
ee - Grade: F
ff - Grade: F
gg - Grade: F
hh - Grade: F
```

Figure 16: Output of Question no: 06