



***School of Mechanical & Manufacturing Engineering (SMME),  
National University of Science and Technology (NUST),  
Sector H-12, Islamabad***

Program: BE-Aerospace

Section: AE-01

Session: Fall 2023

Semester: 1st

Course Title: Fundamentals of Programming (CS-109)

## Assignment

“

”

***Name: Muhammad Amir***

***CMS: 480009***

### **QUESTION.NO.01:**

. 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.

### **ANSWER 01:**

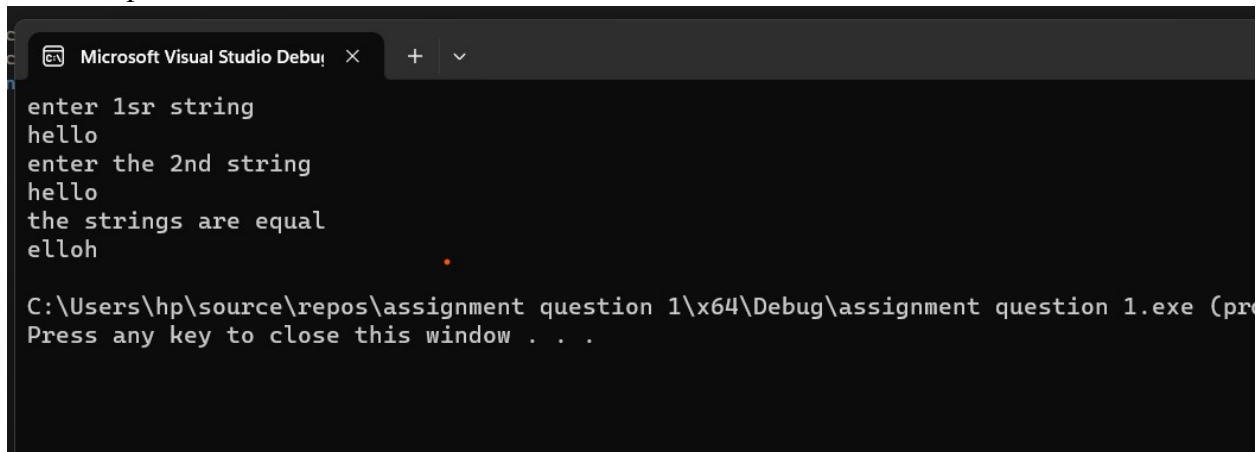
When we run this program Compiler will take two strings as input from the user and compare them to check if they are equal.

If the strings are equal, we proceed to rotate the first string. Here's a breakdown of the rotation process using a for loop:

- We create a copy of the first string called `rotated\_str`.
- We store the first character of `rotated\_str` in a temporary variable `temp`.
- We then use a for loop to iterate through the characters of the string, starting from the first character up to the second-to-last character.
- Inside the for loop, we shift each character one position to the left by assigning the value of the next character to the current character.
- Finally, we assign the value of `temp` to the last character of `rotated\_str`, completing the rotation.

```
#include<iostream>
#include<string>
using namespace std;
int main() {
    string str1, str2;
    cout << "enter 1sr string\n";
    cin >> str1;
    cout << "enter the 2nd string\n";
    cin >> str2;
    if (str1 == str2) {
        string rotated_str = str1;
        char temp = rotated_str[0];
        for (int i = 0; i < str1.length() - 1; i++) {
            rotated_str[i] = rotated_str[i + 1];
        }
        rotated_str[str1.length() - 1] = temp;
        cout << "the strings are equal\n" << rotated_str << endl;
    } else {
        cout << "strings are not equal\n";
    }
    return 0;
}
```

. After the rotation, the compiler will output the rotating string to the console.  
If the strings are not equal, the compiler simply outputs a message indicating that the strings are not equal



```
Microsoft Visual Studio Debug Console
enter 1st string
hello
enter the 2nd string
hello
the strings are equal
elloh

C:\Users\hp\source\repos\assignment question 1\x64\Debug\assignment question 1.exe (pr
Press any key to close this window . . .
```

---

### **QUESTION.NO.02:**

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.

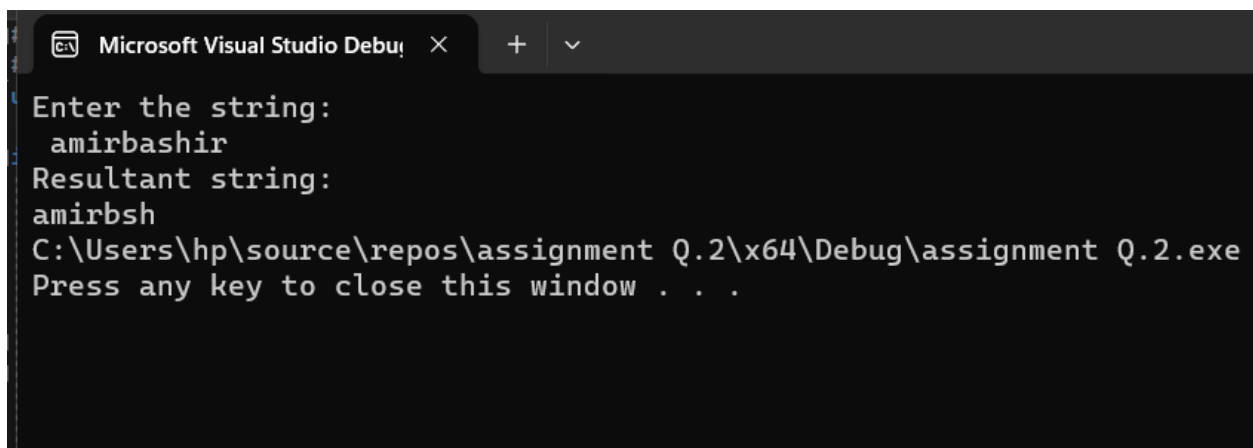
### **ANSWER.02**

- We include the necessary header files: `iostream` for input/output and `string` for string manipulation.
- In the main function, we declare a string variable `str` to store the input string provided by the user.
- We prompt the user to enter the string using the `cout` and `cin` statements.
- We declare an empty string variable `result` to store the resultant string with duplicate characters removed.
- We use a for loop to iterate through each character in the input string `str`.
- Inside the loop, we use the `find` function to check if the current character is already present in the `result` string. If it's not found (i.e., `string::npos`), we append the character to the `result` string.

- After the loop completes, we have the `result` string containing only unique characters from the input string.
- Finally, we print the resultant string using the `cout` statement.

This program effectively removes duplicate characters from the input string and prints the resultant string.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main() {
6      string str;
7      cout << "Enter the string: ";
8      cin >> str;
9
10     string result = "";
11     for (int i = 0; i < str.length(); i++) {
12         if (result.find(str[i]) == string::npos) {
13             result += str[i];
14         }
15     }
16
17     cout << "Resultant string: " << result;
18     return 0;
19 }
20
```



```
Microsoft Visual Studio Debug Console
Enter the string:
amirbashir
Resultant string:
amirbsh
C:\Users\hp\source\repos\assignment Q.2\x64\Debug\assignment Q.2.exe
Press any key to close this window . . .
```

---

### **QUESTION.NO,03:**

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

### Answer 03:

- We include the necessary header file `iostream` to enable input and output operations and use the `using namespace std;` statement to avoid having to write `std::` before standard library functions.
- Inside the main function:
- We declare an integer array `a` with a size of 8 and initialize it with the original values {1, 2, 3, 4, 5}.
- We declare an array `additionalElements` to store the additional user-defined elements.
- We prompt the user to enter 3 additional elements for the array and store them in the `additionalElements` array using a for loop and `cin` for input.
- We then add the additional elements to the original array `a` by iterating through the `additionalElements` array and adding them at positions 5, 6, and 7.
- Finally, we display the elements of the updated array using another for loop and the `cout` statement.
- When the program is executed, it will prompt the user to enter 3 additional elements, then display the updated array elements on the console. program is executed, it will output the elements of the array as "1 2 3 4 5 6 7 8" on the console.

```
t Q.2 (Global Scope)
#include <iostream>
using namespace std;

int main() {
    int a[5] = { 1, 2, 3, 4, 5 };
    int newsz = 8;
    int addEle[] = { 6, 7, 8 };
    int newArray[8];
    for (int i = 0; i < 5; i++) {
        newArray[i] = a[i];
    }
    for (int i = 5; i < newsz; i++) {
        newArray[i] = addEle[i - 5];
    }
    cout << "Combined array elements: ";
    for (int i = 0; i < newsz; i++) {
        cout << newArray[i] << " ";
    }
    cout << endl;
    return 0;
}
```

```
Microsoft Visual Studio Debug Console
Combined array elements: 1 2 3 4 5 6 7 8
C:\Users\hp\source\repos\assignment Q.2\x64\Debug\assignment Q.2.exe
Press any key to close this window . . .
```

---

### **Question.no 04**

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.

### **Answer.04**

- The program takes a positive integer N as input from the user.
- It uses a for loop to iterate through numbers from N downwards.
- For each number, it checks if it is prime by testing divisibility with numbers up to its square root.
- Once the largest prime number is found, it is displayed as the output.

```
#include <iostream>
using namespace std;
int main() {
    int N, i, j;
    bool isPrime;
    cout << "Enter a positive integer N:\n ";
    cin >> N;
    for (i = N; i >= 2; i--) {
        isPrime = true;
        for (j = 2; j * j <= i; j++) {
            if (i % j == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            cout << "The largest prime number less than or equal to N is:\n" << i;
            break;
        }
    }
    return 0;
}
```

```
Microsoft Visual Studio Debug Console
Enter a positive integer N:
66
The largest prime number less than or equal to N is:
61
C:\Users\hp\source\repos\assignment Q.2\x64\Debug\assignment Q.2.exe
Press any key to close this window . . .
```

---

### **Question.no.05;**

Implement Bubble Sort on an array of 6 integers.

### **Answer 05.**

- An array of integers is initialized.
- The bubble sort algorithm compares adjacent elements in the array and swaps them if they are in the wrong order.
- This process is repeated for each element in the array until the entire array is sorted in ascending order.
- The sorted array is then displayed as output. This algorithm is called "bubble sort" because smaller elements gradually "bubble" their way to the top of the array, like bubbles rising in a liquid.

```

#include <iostream>
using namespace std;
int main() {
    int arr[] = { 64, 34, 25, 12, 22, 11 };
    int n = 6;
    int temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    cout << " bubble Sorted array: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    return 0;
}

```

```

#include <iostream>
Microsoft Visual Studio Debug Console
bubble Sorted array: 11 12 22 25 34 64
C:\Users\hp\source\repos\assignment Q.2\x64\Debug\assignment
Press any key to close this window . . .

```

---

### **Question.no.06:**

This is a real life aircraft problem as the system will calculate the remaining distance the aircraft can cover with the current



stored fuel and also warns the pilot as he is about stall or not at the current angle of attack.

- our program begins by initializing variables, including `n` (set to 10), `angleOfAttack`, `capacity`, `fuel`, and `stallAngle`.
- The user is prompted to enter the fuel capacity of the aircraft in liters and the total fuel stored as shown on the meter.
- The user is then asked to set a stall angle for the aircraft.
- Next, the user is prompted to enter the current angle of attack for the aircraft.
- The program checks if the aircraft is near the stall point by comparing the current angle of attack with the stall angle. If the angle of attack is within 5 degrees of the stall angle, a warning message is displayed to reduce the angle of attack immediately. Otherwise, the program informs the user to maintain the current angle of attack.
- Additionally, the program checks the remaining fuel and provides a warning if the fuel level is below 85% of the capacity. If the fuel level is above 85% of the capacity, the program calculates and displays the distance the aircraft can cover with the remaining fuel.
- The program then ends after performing the above checks and calculations.

This program essentially allows the user to input the fuel capacity, current fuel level, stall angle, and angle of attack, and then provides warnings and information based on these inputs.

```

#include <iostream>
using namespace std;
int main() {
    int n=4;
    double angleOfAttack;
    double capacity,fuel;
    double stallAngle;

    //for fuel capacity and remaining fuel
    cout << "enter the fuel capacity of aircraft in liters\n";
    cin >> capacity;
    cout << "total fuel stored showing on meter is in liters \n";
    cin >> fuel;

    // the stall angle from the user
    cout << "set a stall angle: \n";
    cin >> stallAngle;

    //the angle of attack from the user
    cout << "Enter the current angle of attack:\n ";
    cin >> angleOfAttack;

    // Check if the aircraft is near the stall point
    if (angleOfAttack >= (stallAngle - 5) && angleOfAttack <= (stallAngle + 5)) {
        cout << "You are near the stall point. Reduce the angle of attack immediately." << endl;
    }
    else {
        cout << "You are not near the stall point. Maintain your current angle of attack." << endl;
    }

    // for remaining fuel and the distance the aircraft can cover
    if (fuel <= (capacity - capacity * 85 / 100)) {
        cout << "low fuel warning\n";
    }
    else {
        cout << "with this fuel you can fly \n";
        cout << fuel / n << " " << "kilometers\n";
    }

    return 0;
}

```

The out put of our program is as

```

Microsoft Visual Studio Debug Console
+ v

enter the fuel capacity of aircraft in liters
770
total fuel stored showing on meter is in liters
550
set a stall angle:
56
Enter the current angle of attack:
54
You are near the stall point. Reduce the angle of attack immediately.
with this fuel you can fly
137.5 kilometers

C:\Users\hp\source\repos\assignment Q.2\x64\Debug\assignment Q.2.exe (process)
Press any key to close this window . . .

```