A blue and white logo

Description automatically generated

**NUST SCHOOL OF MECHANICAL AND MANUFACTURING ENGINEERING, SMME**

**FUNDAMENTALS OF PROGRAMMING**

**ASSIGNMENT#01**

Section:AE-01

Course Code: CS-109

Department of Aerospace Engineering

Submitted to:

Ma’am Laiba

Submitted by:

Khansa Malik

CMS ID:

453832

**(Question#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.

**EXPLANATION:**

This code compares two strings entered by the user and checks if they are equal. If they are equal, it rotates the first string by moving each character one position to the left and places the first character at the end. Here's how it works:

1. The code includes the necessary header files for input/output and string manipulation.
2. The **main** function is the entry point of the program.
3. It declares two string variables **str1** and **str2** to store the user input.
4. The code prompts the user to enter the first string using the **cout** object and reads the input using the **cin** object.
5. It then prompts the user to enter the second string and reads the input.
6. The code compares the two strings using the **==** operator.
7. If the strings are equal, it enters the **if** block.
8. Inside the **if** block, it stores the first character of **str1** in the variable **firstChar**.
9. It then uses a **for** loop to iterate over the characters of **str1** starting from the first character.
10. In each iteration, it assigns the value of the next character to the current character, effectively shifting all characters one position to the left.
11. Finally, it places the first character (**firstChar**) at the end of **str1** to complete the rotation.
12. It prints the rotated string to the console using the **cout** object.
13. If the strings are not equal, it enters the **else** block and prints a message to the console.
14. The program then returns 0 to indicate successful execution.

**CODE:**

#include <iostream>

#include <string>

using namespace std;

int main()

{

string str1, str2;

cout << "Enter the first string: ";

cin >> str1;

cout << "Enter the second string: ";

cin >> str2;

if (str1 == str2)

{

char firstChar = str1[0];

for (int i = 0; i < str1.length() - 1; i++)

{

str1[i] = str1[i + 1];

}

str1[str1.length() - 1] = firstChar;

cout << "Both strings are equal and rotated 1st string is: " << str1 << endl;

}

else

{

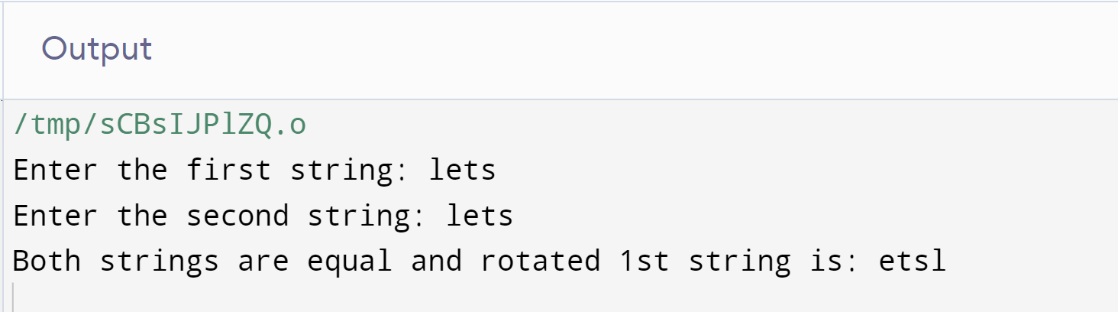
cout<<"Strings are not equal."<< endl;

}

return 0;

}

**OUTPUT:**



**(Question#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.

**EXPLANATION:**

This code takes a string as input from the user and removes any duplicate characters from it. Here's how it works:

1. The code includes the necessary header files for input/output and string manipulation.
2. The **main** function is the entry point of the program.
3. It declares a string variable **str** to store the user input.
4. It prompts the user to enter a string using the **cout** object and reads the input using the **cin** object.
5. The code then iterates over each character of the string using a **for** loop
6. For each character, it checks if there are any duplicate occurrences of that character in the rest of the string.
7. If a duplicate character is found, it uses the erase function to remove it from the string.
8. If no duplicate character is found, the inner for loop proceeds to the next character.
9. After all duplicates have been removed, the resulting string is printed to the console using the cout object.

**CODE:**

#include <iostream>

#include <string>

using namespace std;

int main()

{

string str;

cout << "Enter a string: ";

cin >> str;

for (int i = 0; i < str.length(); ++i)

{

char x = str[i];

for (int j = i + 1; j < str.length();)

{

if (str[j] == x)

{

str.erase(j, 1);

}

else

{

++j;

}

}

}

cout << "Resultant string after removing duplicates: " << str << endl;

return 0;

}

**OUTPUT:**

A screenshot of a computer

Description automatically generated

**(QUESTION#03)**

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

**EXPLANATION:**

This code creates an array of integers with a predefined size of 5 and allows the user to add additional elements to it. Here's how it works:

1. The code includes the necessary header file for input/output.
2. The **main** function is the entry point of the program.
3. It declares an integer array **a** with a size of 5 and initializes it with values 1, 2, 3, 4, and 5.
4. The code then prints the original array to the console using the **cout** object and a **for** loop.
5. It prompts the user to enter the number of additional elements they want to add to the array using the **cout** object and reads the input using the **cin** object, storing it in the variable **n**.
6. The program calculates the new size of the array by adding the number of additional elements entered by the user to the original size (5).
7. It creates a new array **newArray** with the new size.
8. The code then uses a **for** loop to copy the elements from the original array **a** to the new array **newArray**.
9. After copying the original elements, the program uses another **for** loop to prompt the user to enter the additional elements one by one, starting from the index 5 of the new array. The entered elements are then stored in the new array.
10. Finally, the program prints the elements in the new array to the console using the **cout** object and a **for** loop.
11. The program then returns 0 to indicate successful execution.

**CODE:**

#include <iostream>

using namespace std;

int main()

{

int a[5] = {1, 2, 3, 4, 5};

int n;

cout << "Original array: ";

for (int i = 0; i < 5; ++i)

{

cout << a[i] << " ";

}

cout << endl;

cout << "Enter the number of additional elements: ";

cin >> n;

int newSize = 5 + n;

int newArray[newSize];

for (int i = 0; i < 5; i++) {

newArray[i] = a[i];

}

for (int i = 5; i < newSize; i++) {

cout << "Enter element " << (i - 4) << ": ";

cin >> newArray[i];

}

cout << "Elements in the array: ";

for (int i = 0; i < newSize; i++) {

cout << newArray[i] << " ";

}

return 0;

}

**OUTPUT:**

A screenshot of a computer code

Description automatically generated

**(Question#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.

**EXPLANATION:**

This code finds the largest prime number less than or equal to a positive integer entered by the user. Here's how it works:

1. The code includes the necessary header file for input/output.
2. The **main** function is the entry point of the program.
3. It declares an integer variable **num** to store the user input.
4. The code prompts the user to enter a positive integer using the **cout** object and reads the input using the **cin** object, storing it in the variable **num**.
5. It initializes the variable **largestPrime** to **num - 1**, assuming that the largest prime number is initially one less than the entered number.
6. The code enters a **while** loop that continues until a prime number is found or **largestPrime** becomes less than 2.
7. Within the loop, it assumes that **largestPrime** is prime by setting the boolean variable **isPrime** to **true**.
8. It then uses a **for** loop to check if **largestPrime** is divisible by any number from 2 to the square root of **largestPrime**.
9. If **largestPrime** is divisible by any number, it sets **isPrime** to **false** and breaks out of the loop.
10. If **largestPrime** is not divisible by any number, it remains **true**, indicating that it is prime.
11. If **largestPrime** is prime, it exits the **while** loop.
12. If **largestPrime** is not prime, it decrements **largestPrime** by 1 and continues the loop.
13. After exiting the loop, the code prints the largest prime number less than or equal to the positive integer to the console using the **cout** object.
14. Finally, the program returns 0 to indicate successful execution.

**CODE:**

#include <iostream>

using namespace std;

int main() {

int num;

cout << "Enter a positive integer: ";

cin >> num;

int largestPrime = num - 1;

while (largestPrime >= 2)

{

bool isPrime = true;

for (int i = 2; i \* i <= largestPrime; i++)

{

if (largestPrime % i == 0)

{

isPrime = false;

break;

}

}

if (isPrime)

{

break;

}

largestPrime--;

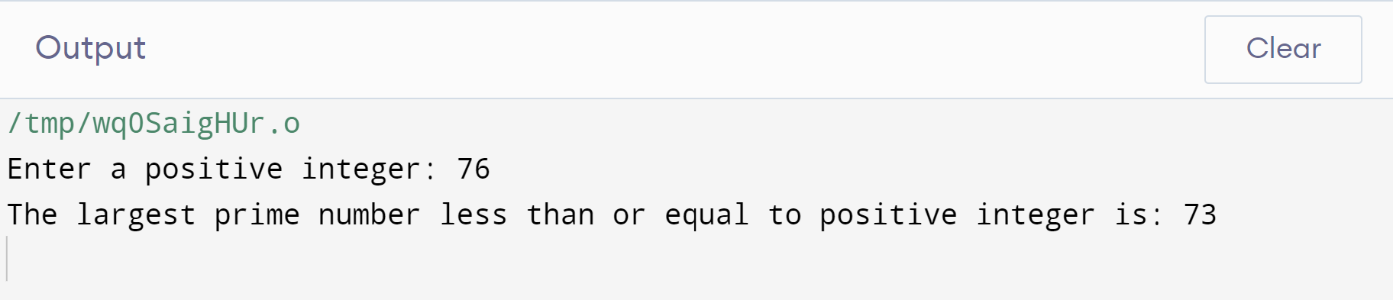
}

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

return 0;

}

**OUTPUT:**



**(Question#05)**

Implement Bubble Sort on an array of 6 integers.

**EXPLANATION:**

This code takes 6 integers as input from the user, sorts them in ascending order using the bubble sort algorithm, and then prints the sorted array. Here's how it works:

1. The code includes the necessary header file for input/output.
2. The **main** function is the entry point of the program.
3. It declares an integer array **a** of size 6 to store the input integers.
4. The code prompts the user to enter 6 integers using the **cout** object.
5. It reads the integers into the array **a** using the **cin** object and a **for** loop.
6. The program then performs the bubble sort algorithm to sort the array **a** in ascending order.
7. It uses two nested **for** loops to compare adjacent elements of the array and swap them if they are in the wrong order.
8. The variable **tmp** is used as a temporary variable to hold the value of an element while swapping.
9. After sorting the array, the program prints the sorted array to the console using the **cout** object and a **for** loop.
10. Finally, the program returns 0 to indicate successful execution.

**CODE:**

#include <iostream>

using namespace std;

int main()

{

int a[6];

cout << "Enter 6 integers: ";

for (int n = 0; n < 6; n++)

{

cin >> a[n];

}

int tmp;

for (int i=0; i<5 ; i++)

{

for (int j=0; j<=5-i;j++)

{

if (a[j]>a[j+1])

{

tmp=a[j];

a[j]=a[j+1];

a[j+1]=tmp;

}

}

}

for(int k=0; k<6; k++)

{

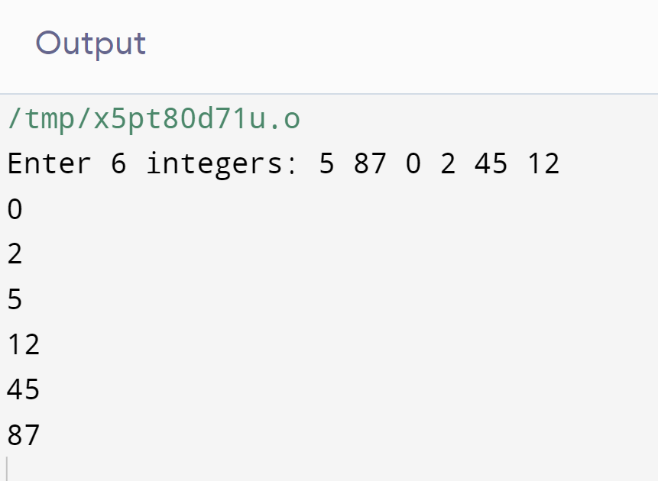
cout<<a[k]<<endl;

}

return 0;

}

**OUTPUT:**



**(Question#06)**

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

**EXPLANATION:**

This code calculates the estimated fuel consumption of an aircraft based on the thrust and specific fuel consumption provided by the user. Here's how it works:

1. The code includes the necessary header file for input/output.
2. The **calculateFuelConsumption** function is defined to calculate the fuel consumption based on the thrust and specific fuel consumption.
3. The function takes in two double parameters: **thrust** (in Newtons) and **specificFuelConsumption** (in kg/N).
4. Inside the function, it multiplies the **thrust** by the **specificFuelConsumption** and returns the result.
5. The **main** function is the entry point of the program.
6. It declares two double variables: **thrust** to store the thrust of the aircraft and **specificFuelConsumption** to store the specific fuel consumption.
7. The code prompts the user to enter the thrust and specific fuel consumption using the **cout** object.
8. It reads the user inputs and stores them in the respective variables using the **cin** object.
9. It then calls the **calculateFuelConsumption** function, passing the **thrust** and **specificFuelConsumption** as arguments, and stores the result in the **fuelConsumption** variable.
10. Finally, the code prints the estimated fuel consumption to the console using the **cout** object.Bottom of Form

**CODE:**

#include <iostream>

using namespace std;

double calculateFuelConsumption(double thrust, double specificFuelConsumption)

{

return thrust \* specificFuelConsumption;

}

int main()

{

// Thrust of the aircraft (N)

double thrust;

// Specific fuel consumption of the aircraft (kg/N)

double specificFuelConsumption;

cout << "Aircraft fuel consumption estimation:\n";

cout << "Enter the thrust (N): ";

cin >> thrust;

cout << "Enter the specific fuel consumption (kg/N): ";

cin >> specificFuelConsumption;

double fuelConsumption = calculateFuelConsumption(thrust, specificFuelConsumption);

cout << "Estimated fuel consumption: " << fuelConsumption << " kg\n";

return 0;

}

**OUTPUT:**

