******

***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)

***“FOP ASSIGNMENT”***

***Name: Hadia Khan***

***CMS:*** ***455885***

**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.

Answer: The program is:

#include <iostream>

#include<string>

#include<algorithm>

using namespace std;

int main()

{

//Input two strings from the user

string str1,str2;

cout<<"Enter the first string: ";

cin>>str1;

cout<<"Enter the second string: ";

cin>>str2;

//Check if the strings are equal

if(str1==str2)

{

cout<<"The string are equal.\n";

rotate(str2.begin(),str2.begin()+1, str2.end());

cout<<"After rotation, the second string is: "<<str2<<"\n";

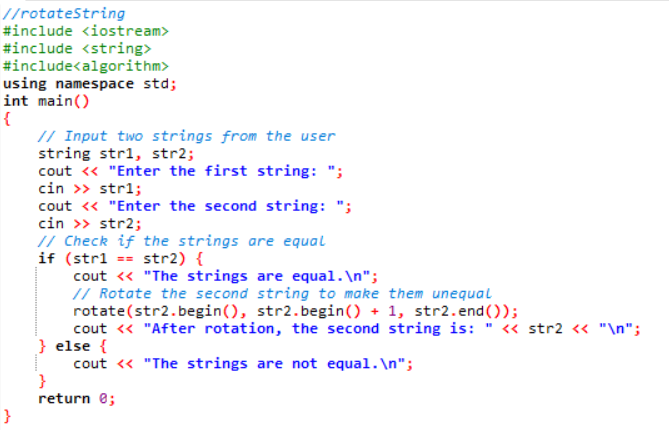
}else{

cout<<"The strings are not equal.\n";

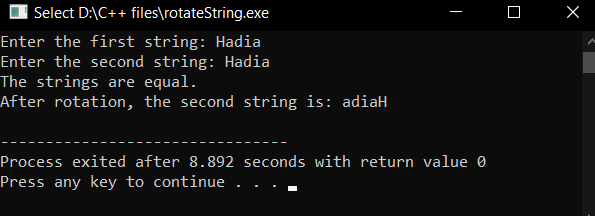
}

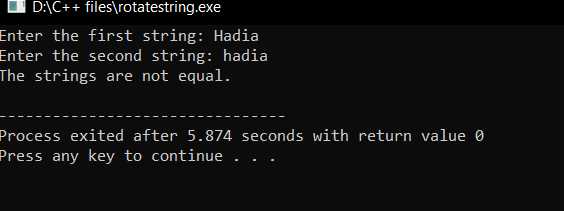
return 0;

}



Outputs:





**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.

Answer: The program is:

#include <iostream>

#include <string>

#include <unordered\_set>

using namespace std;

string removeDuplicates(const string& str) {

unordered\_set<char> seenChars;

string result;

for (char ch : str) {

if (seenChars.insert(ch).second) {

result += ch;

}

}

return result;

}

int main() {

string originalStr;

cout << "Enter a string: ";

getline(cin, originalStr);

string uniqueStr = removeDuplicates(originalStr);

cout << "String without duplicates: " << uniqueStr << endl;

return 0;

}



**QUESTION 03:**

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

Answer: The program is:

//addition of elements in an existing array

#include<iostream>

using namespace std;

int main() {

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

int newSize = 8; // Set the new size of the array

// Creating a new array with a larger size

int newArray[newSize];

// Copying existing elements to the new array

for (int i = 0; i < sizeof(a) / sizeof(a[0]); ++i) {

newArray[i] = a[i];

}

// Adding more elements to the array

for (int i = sizeof(a) / sizeof(a[0]); i < newSize; ++i) {

newArray[i] = i + 1;

}

// Displays the elements

cout << "Array elements: ";

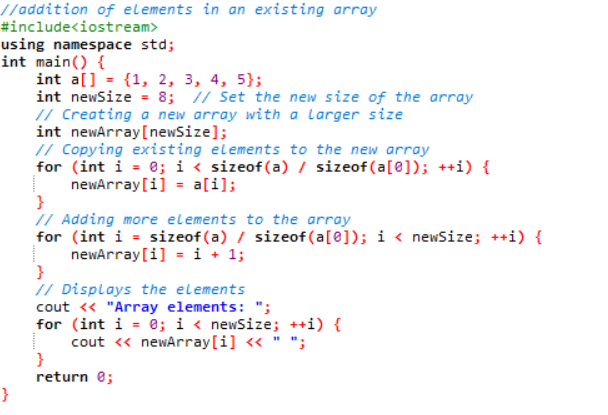
for (int i = 0; i < newSize; ++i) {

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

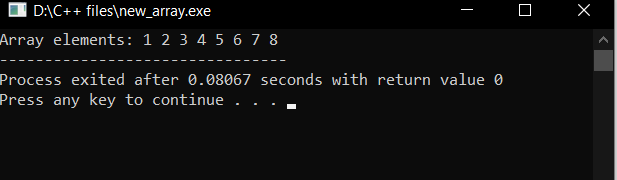
}

return 0;

}



Output:



**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.

Answer: The program is:

#include<iostream>

using namespace std;

bool isPrime(int number) {

if (number < 2) {

return false;

}

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

if (number % i == 0) {

return false;

}

}

return true;

}

int main() {

int N;

// Input the value of N

cout<<"Enter a positive integer N: ";

cin>>N;

// Find the largest prime number less than or equal to N

while (N > 1 && !isPrime(N)) {

--N;

}

// Display the result

if (N > 1) {

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

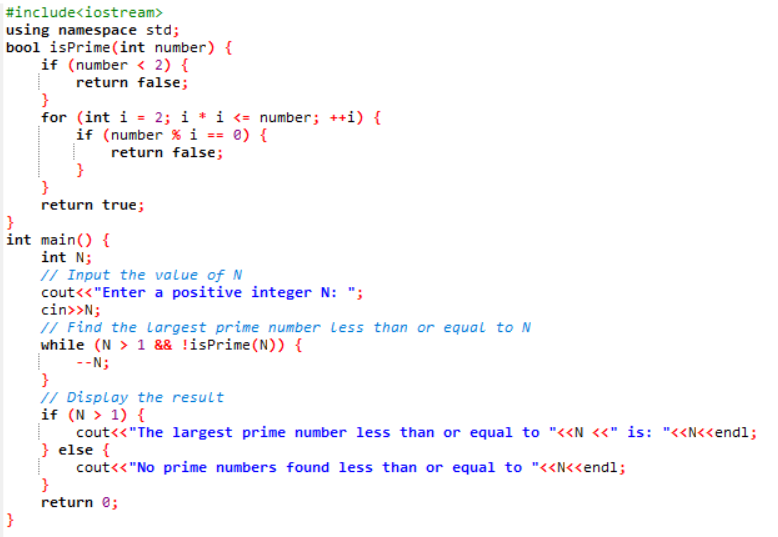
} else {

cout<<"No prime numbers found less than or equal to "<<N<<endl;

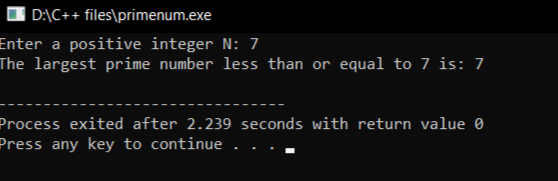
}

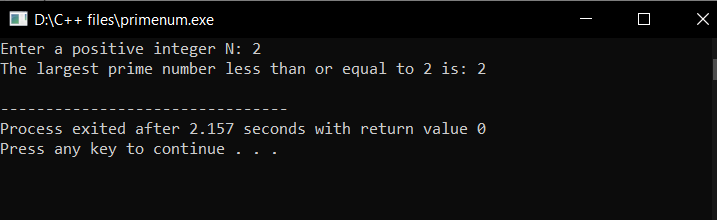
return 0;

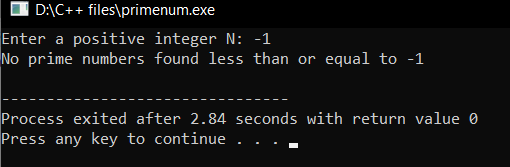
}



Outputs:







**QUESTION 05:**

Implement Bubble Sort on an array of 6 integers.

Answer:

The program is:

//This program is designed to implement the bubble sort on an array

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n)

{

for (int i = 0; i < n - 1; ++i) {

for (int j = 0; j < n - i - 1; ++j)

{

if (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j + 1]

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int main()

{

const int size = 6;

int arr[size] = {5, 2, -9, 1, -1, 6};

cout << "Original Array: ";

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

cout << arr[i] << " ";

}

// Perform Bubble Sort

bubbleSort(arr, size);

cout << "\nSorted Array: ";

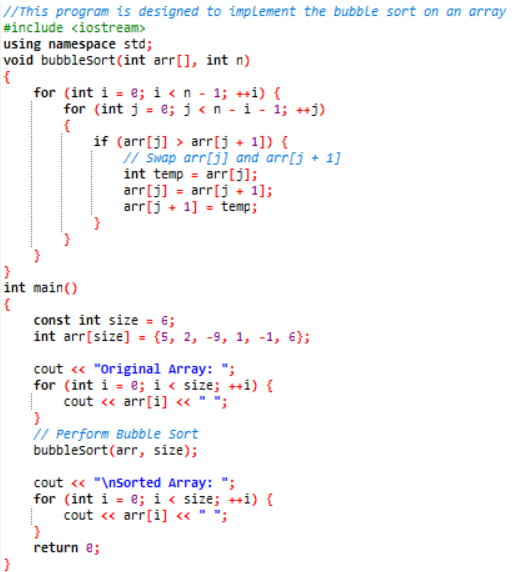
for (int i = 0; i < size; ++i) {

cout << arr[i] << " ";

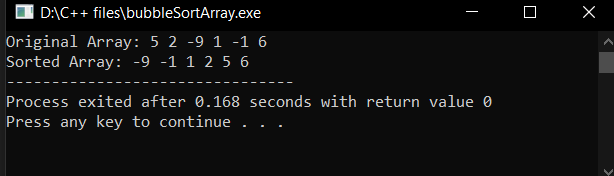
}

return 0;

}



Output:



**QUESTION 06:**

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

Answer: The program is:

//Aerospace\_problem

#include <iostream>

#include <cmath>

using namespace std;

// Function to calculate projectile trajectory

void calculateProjectileTrajectory(double initialVelocity, double launchAngle) {

const double g = 9.81; // Acceleration due to gravity (m/s^2)

// Convert launch angle to radians

double launchAngleRad = launchAngle \* M\_PI / 180.0;

// Calculate time of flight

double timeOfFlight = (2.0 \* initialVelocity \* sin(launchAngleRad)) / g;

// Calculate range

double range = initialVelocity \* cos(launchAngleRad) \* timeOfFlight;

// Calculate maximum height

double maxHeight = pow(initialVelocity \* sin(launchAngleRad), 2.0) / (2.0 \* g);

// Output results

cout << "Projectile Trajectory Calculation\n";

cout << "Initial Velocity: " << initialVelocity << " m/s\n";

cout << "Launch Angle: " << launchAngle << " degrees\n";

cout << "Time of Flight: " << timeOfFlight << " seconds\n";

cout << "Range: " << range << " meters\n";

cout << "Maximum Height: " << maxHeight << " meters\n";

}

int main() {

// Input parameters

double initialVelocity = 100.0; // Initial velocity in m/s

double launchAngle = 45.0; // Launch angle in degrees

// Calculates and displays projectile trajectory

calculateProjectileTrajectory(initialVelocity, launchAngle);

return 0;

}



Output:

