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:1

Name: Bushra Hussain Maqpoon

CMS: 455173

School of Mechanical & Manufacturing Engineering (SMME)

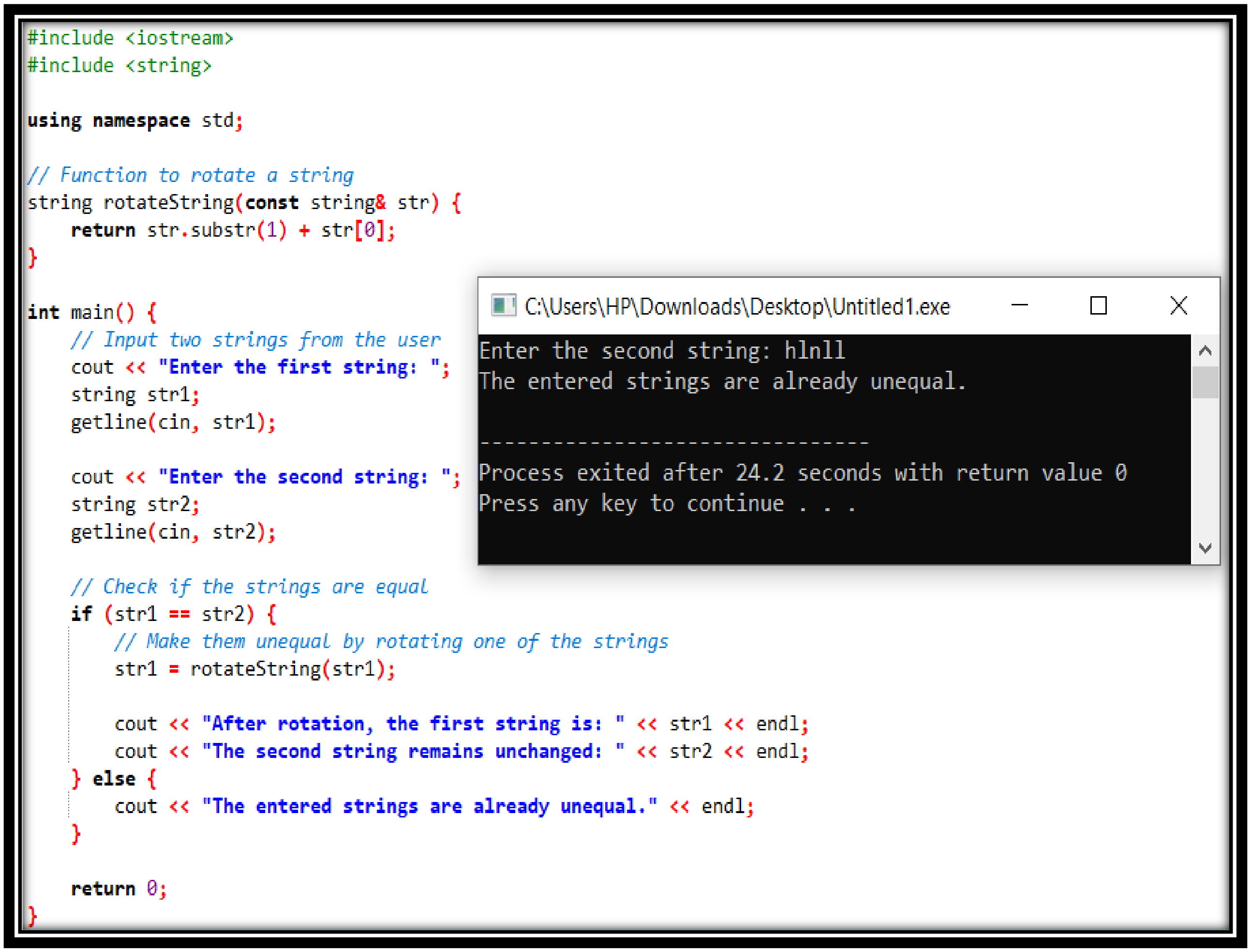
National University of Science and Technology (NUST)

Assignment #01

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



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

using namespace std;

// Function to rotate a string
string rotateString(const string& str) {
    return str.substr(1) + str[0];
}

int main() {
    // Input two strings from the user
    cout << "Enter the first string: ";
    string str1;
    getline(cin, str1);

    cout << "Enter the second string: ";
    string str2;
    getline(cin, str2);

    // Check if the strings are equal
    if (str1 == str2) {
        // Make them unequal by rotating one of the strings
        str1 = rotateString(str1);

        cout << "After rotation, the first string is: " << str1 << endl;
        cout << "The second string remains unchanged: " << str2 << endl;
    } else {
        cout << "The entered strings are already unequal." << endl;
    }

    return 0;
}
```

```
Enter the second string: hlnll
The entered strings are already unequal.

--------------------------------
Process exited after 24.2 seconds with return value 0
Press any key to continue . . .
```

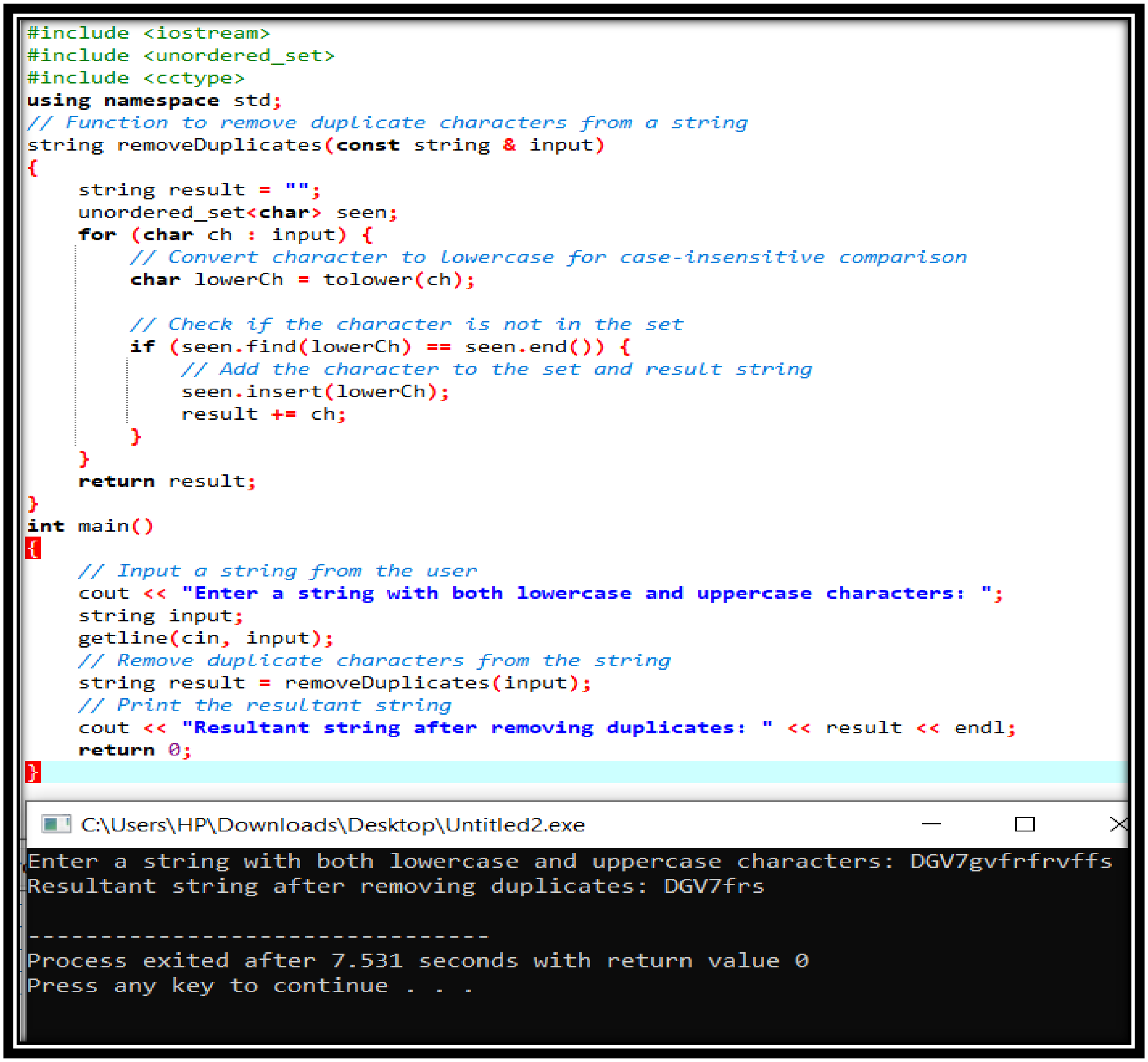
Description:

This program uses the rotateString function to rotate the first string, making it unequal to the second string if they were initially equal. The rotated string is then printed, along with the unchanged second string

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



```cpp
#include <iostream>
#include <unordered_set>
#include <cctype>
using namespace std;
// Function to remove duplicate characters from a string
string removeDuplicates(const string & input)
{
    string result = "";
    unordered_set<char> seen;
    for (char ch : input) {
        // Convert character to lowercase for case-insensitive comparison
        char lowerCh = tolower(ch);

        // Check if the character is not in the set
        if (seen.find(lowerCh) == seen.end()) {
            // Add the character to the set and result string
            seen.insert(lowerCh);
            result += ch;
        }
    }
    return result;
}
int main()
{
    // Input a string from the user
    cout << "Enter a string with both lowercase and uppercase characters: ";
    string input;
    getline(cin, input);
    // Remove duplicate characters from the string
    string result = removeDuplicates(input);
    // Print the resultant string
    cout << "Resultant string after removing duplicates: " << result << endl;
    return 0;
}
```

```
Enter a string with both lowercase and uppercase characters: DGV7gvfrfrvffs
Resultant string after removing duplicates: DGV7frs

--------------------------------
Process exited after 7.531 seconds with return value 0
Press any key to continue . . .
```

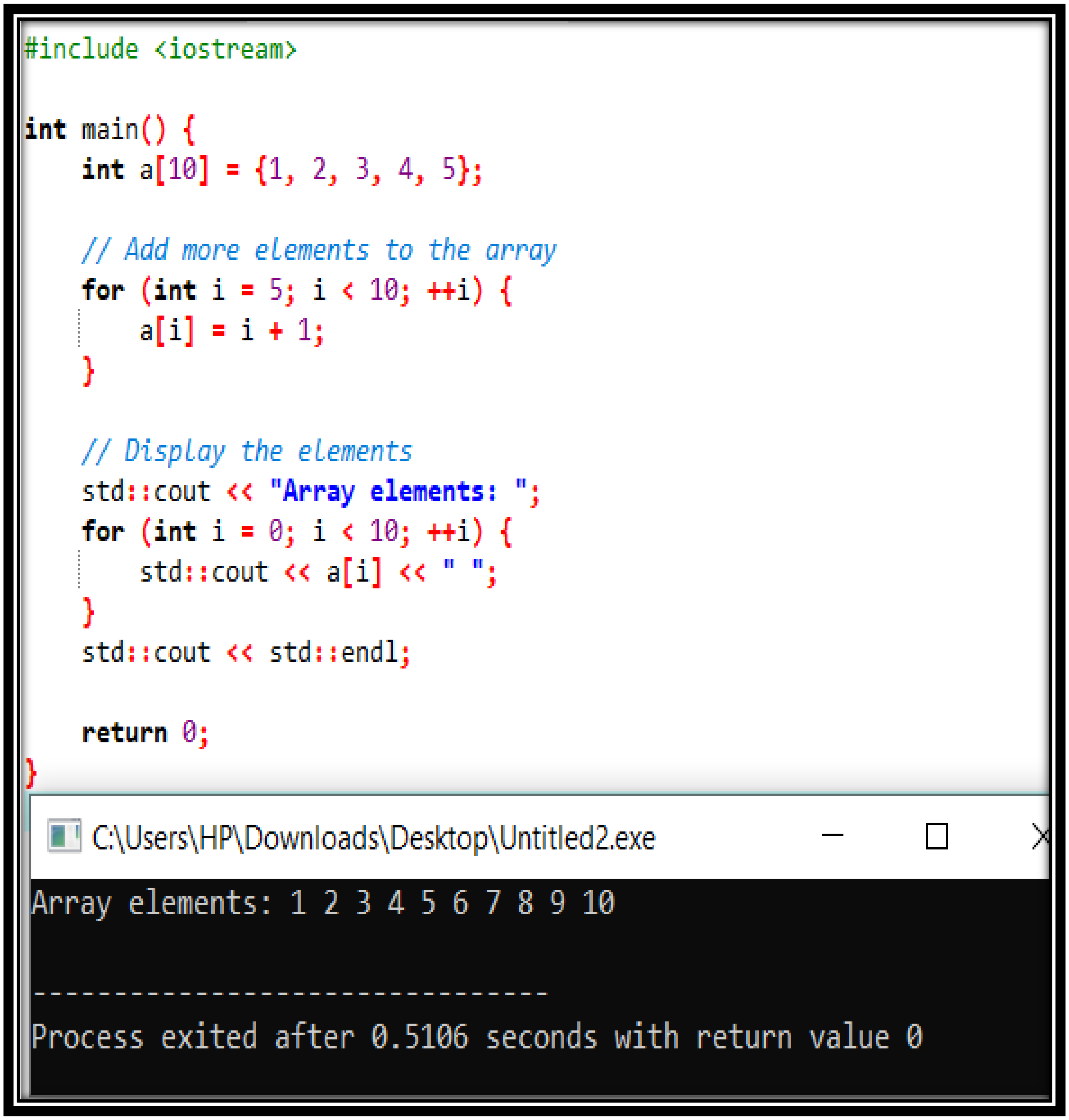
Description:

An unordered set is used by this program to record the characters that are viewed when it traverses the input string. To make the comparison case-insensitive, it first changes every character to lowercase (using tolower) before looking for duplicates. Characters that are not already in the set are added to create the final string.

Question #03

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

Answer



```cpp
#include <iostream>

int main() {
    int a[10] = {1, 2, 3, 4, 5};

    // Add more elements to the array
    for (int i = 5; i < 10; ++i) {
        a[i] = i + 1;
    }

    // Display the elements
    std::cout << "Array elements: ";
    for (int i = 0; i < 10; ++i) {
        std::cout << a[i] << " ";
    }
    std::cout << std::endl;

    return 0;
}
```

```
Array elements: 1 2 3 4 5 6 7 8 9 10

--------------------------------
Process exited after 0.5106 seconds with return value 0
```

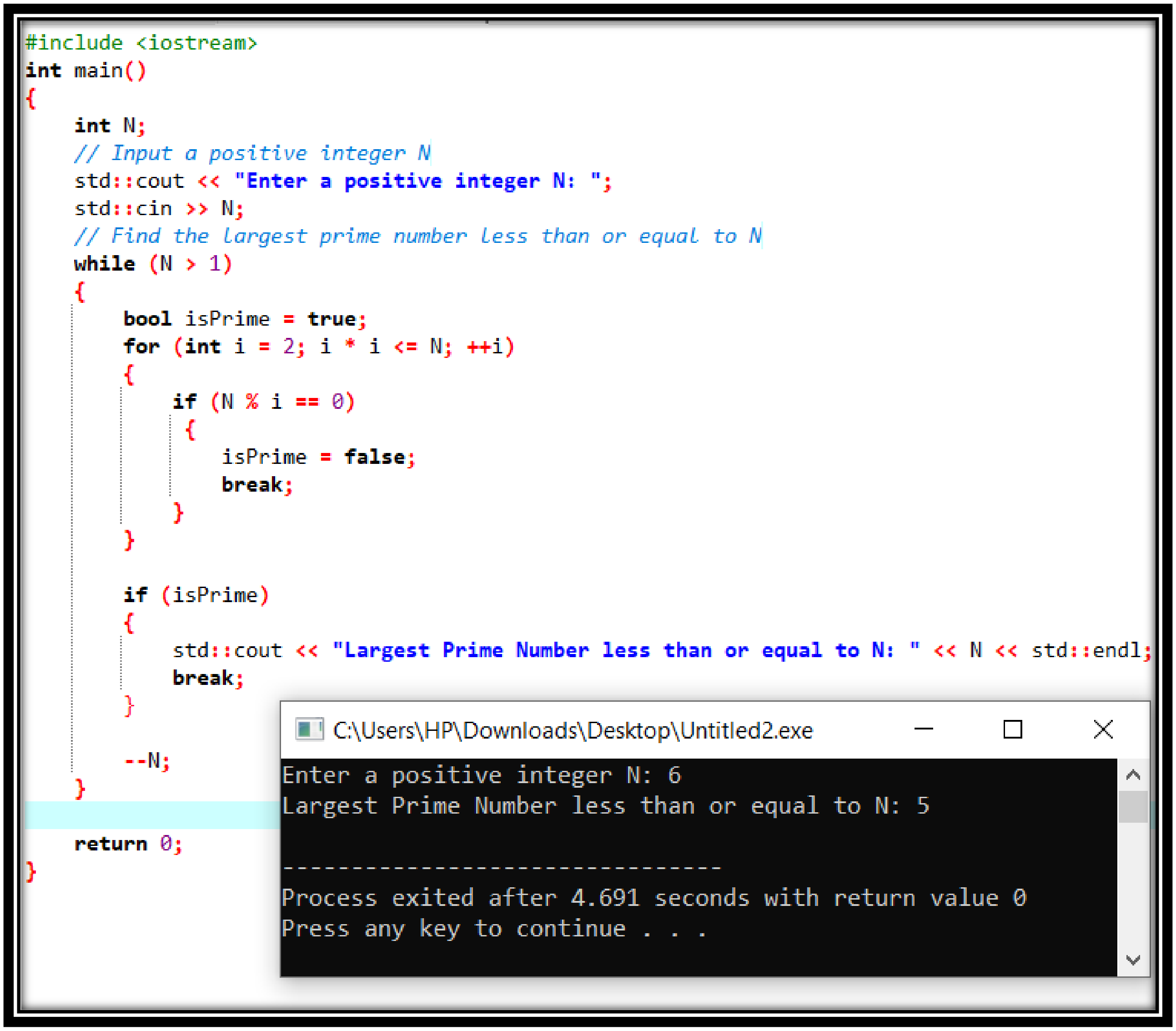
Description

This code adds elements 6 through 10 to the array and then prints all 10 elements. Just note that the array is now of size 10, and elements 6 to 10 are initialized in the loop. If you want to make the array dynamic and add elements as needed.

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



```cpp
#include <iostream>
int main()
{
    int N;
    // Input a positive integer N
    std::cout << "Enter a positive integer N: ";
    std::cin >> N;
    // Find the largest prime number less than or equal to N
    while (N > 1)
    {
        bool isPrime = true;
        for (int i = 2; i * i <= N; ++i)
        {
            if (N % i == 0)
            {
                isPrime = false;
                break;
            }
        }

        if (isPrime)
        {
            std::cout << "Largest Prime Number less than or equal to N: " << N << std::endl;
            break;
        }

        --N;
    }

    return 0;
}
```

```
Enter a positive integer N: 6
Largest Prime Number less than or equal to N: 5


--------------------------------
Process exited after 4.691 seconds with return value 0
Press any key to continue . . .
```

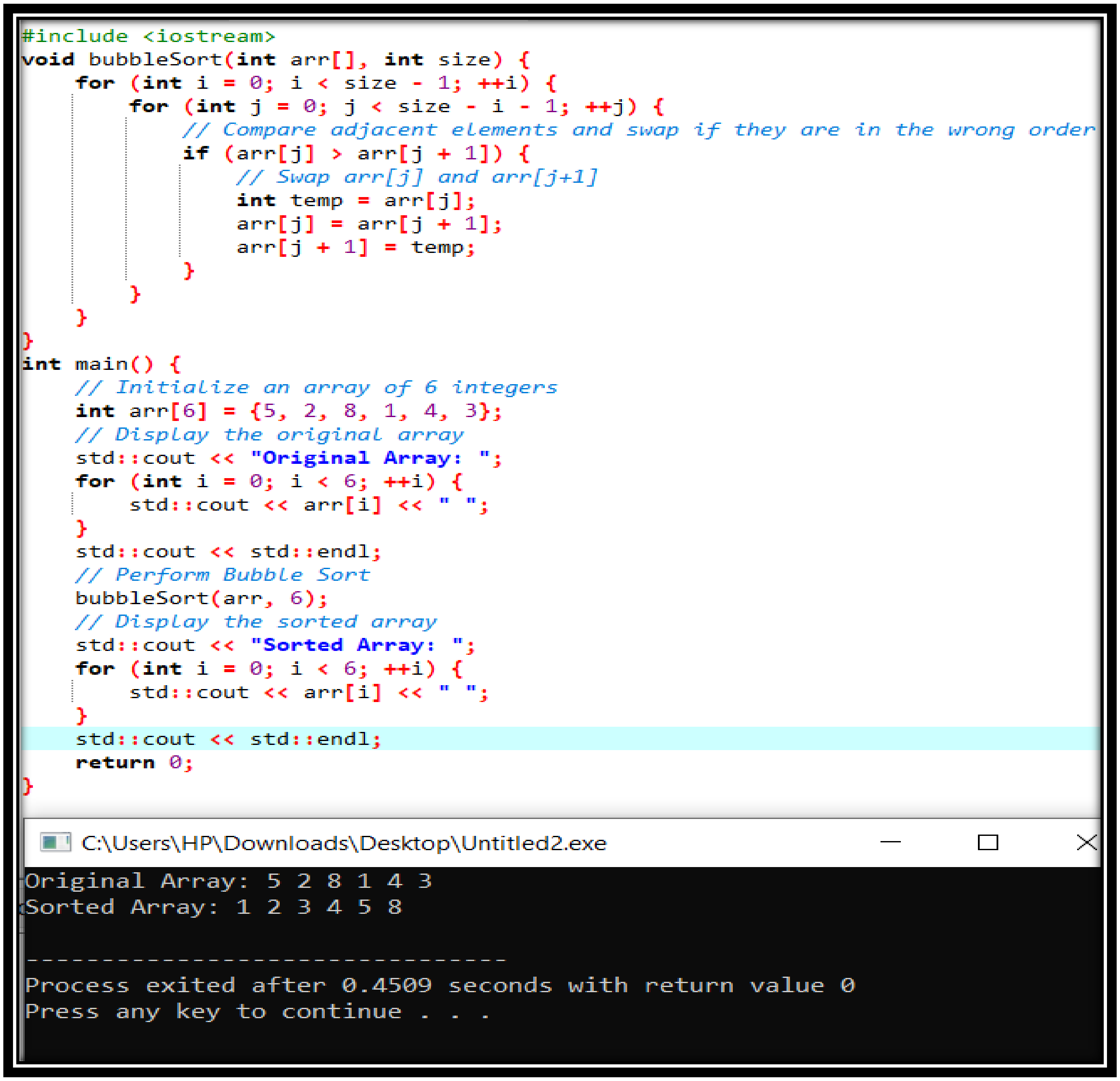
Description

This C++ program takes a positive integer N as user input and efficiently finds the largest prime number less than or equal to N using a while loop. It decrements N, checks for primality, and prints the result, terminating after identifying the largest prime number. The program avoids user-defined functions for simplicity

Question #05

Implement Bubble Sort on an array of 6 integers

Answer



```cpp
#include <iostream>
void bubbleSort(int arr[], int size) {
    for (int i = 0; i < size - 1; ++i) {
        for (int j = 0; j < size - i - 1; ++j) {
            // Compare adjacent elements and swap if they are in the wrong order
            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() {
    // Initialize an array of 6 integers
    int arr[6] = {5, 2, 8, 1, 4, 3};
    // Display the original array
    std::cout << "Original Array: ";
    for (int i = 0; i < 6; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
    // Perform Bubble Sort
    bubbleSort(arr, 6);
    // Display the sorted array
    std::cout << "Sorted Array: ";
    for (int i = 0; i < 6; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
    return 0;
}
```

```
Original Array: 5 2 8 1 4 3
Sorted Array: 1 2 3 4 5 8

--------------------------------
Process exited after 0.4509 seconds with return value 0
Press any key to continue . . .
```

Description

This program defines a bubbleSort function to perform the Bubble Sort algorithm on an array. The main function initializes an array of 6 integers, displays the original array, performs the sorting using Bubble Sort, and then displays the sorted array.

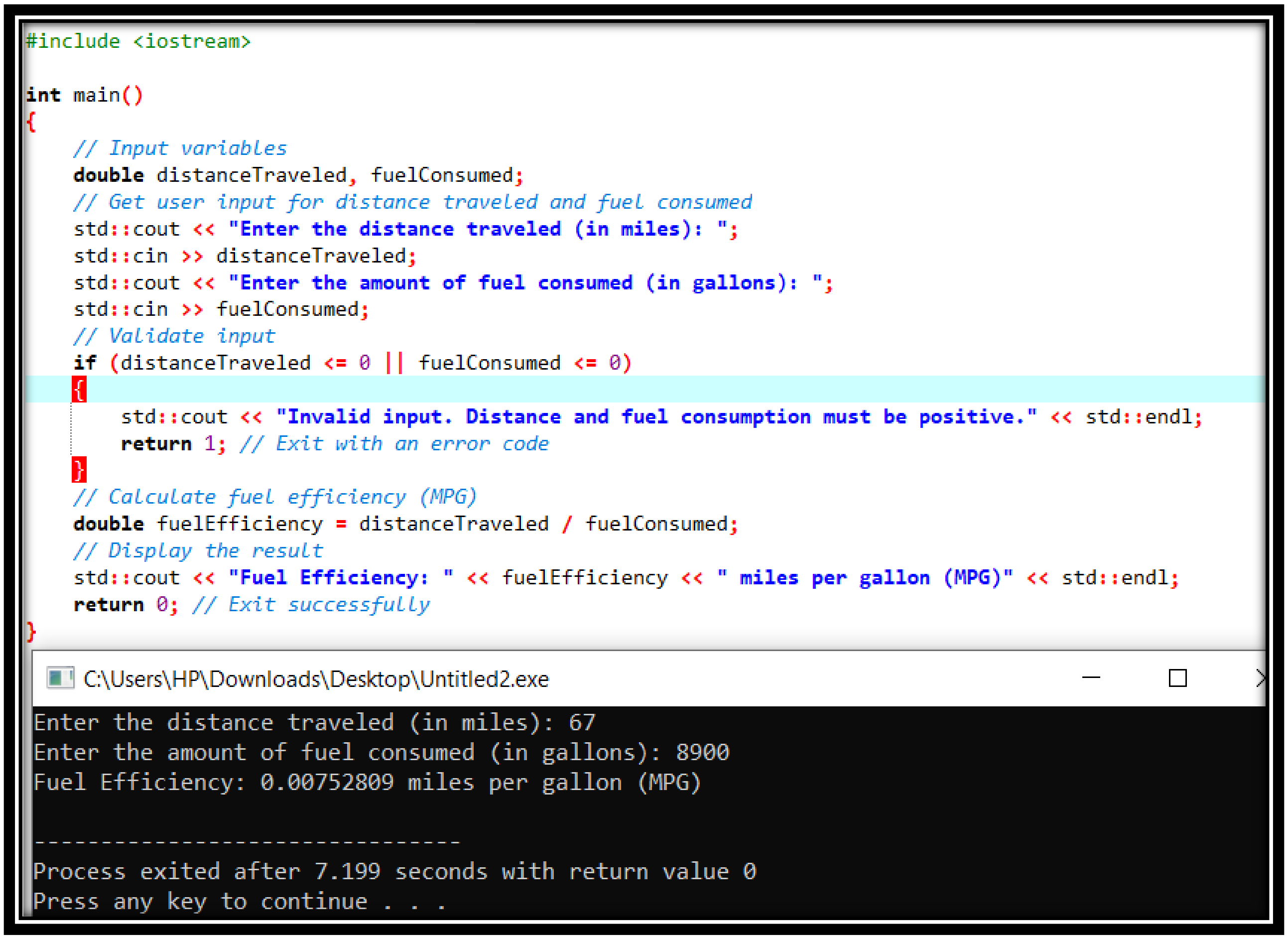
Question #06

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

Answer

Let's look at a straightforward real-world aerospace engineering problem: figuring out an aircraft's fuel economy.

Problem statement: Determine and show the fuel economy in miles per gallon (MPG) based on the aircraft's distance flown and fuel consumption.



```cpp
#include <iostream>

int main()
{
    // Input variables
    double distanceTraveled, fuelConsumed;
    // Get user input for distance traveled and fuel consumed
    std::cout << "Enter the distance traveled (in miles): ";
    std::cin >> distanceTraveled;
    std::cout << "Enter the amount of fuel consumed (in gallons): ";
    std::cin >> fuelConsumed;
    // Validate input
    if (distanceTraveled <= 0 || fuelConsumed <= 0)
    {
        std::cout << "Invalid input. Distance and fuel consumption must be positive." << std::endl;
        return 1; // Exit with an error code
    }
    // Calculate fuel efficiency (MPG)
    double fuelEfficiency = distanceTraveled / fuelConsumed;
    // Display the result
    std::cout << "Fuel Efficiency: " << fuelEfficiency << " miles per gallon (MPG)" << std::endl;
    return 0; // Exit successfully
}
```

```
Enter the distance traveled (in miles): 67
Enter the amount of fuel consumed (in gallons): 8900
Fuel Efficiency: 0.00752809 miles per gallon (MPG)

--------------------------------
Process exited after 7.199 seconds with return value 0
Press any key to continue . . .
```

Description

In this C++ program, the user is prompted to input the distance traveled and the amount of fuel consumed by an aircraft. The program then calculates the fuel efficiency in terms of miles per gallon (MPG) using the formula:

MPG = distanceTraveled / fuelConsumed. The result is then displayed to the user.This problem is a simplified representation of a common scenario in aerospace engineering, where understanding the fuel efficiency of an aircraft is crucial for optimizing performance and managing resources.