Assignment # 1

Name: Rumman Tanvir
CMS: 482064
Program: BE-Aerospace
Session: Fall 2023
Section: AE-01
Semester: 1

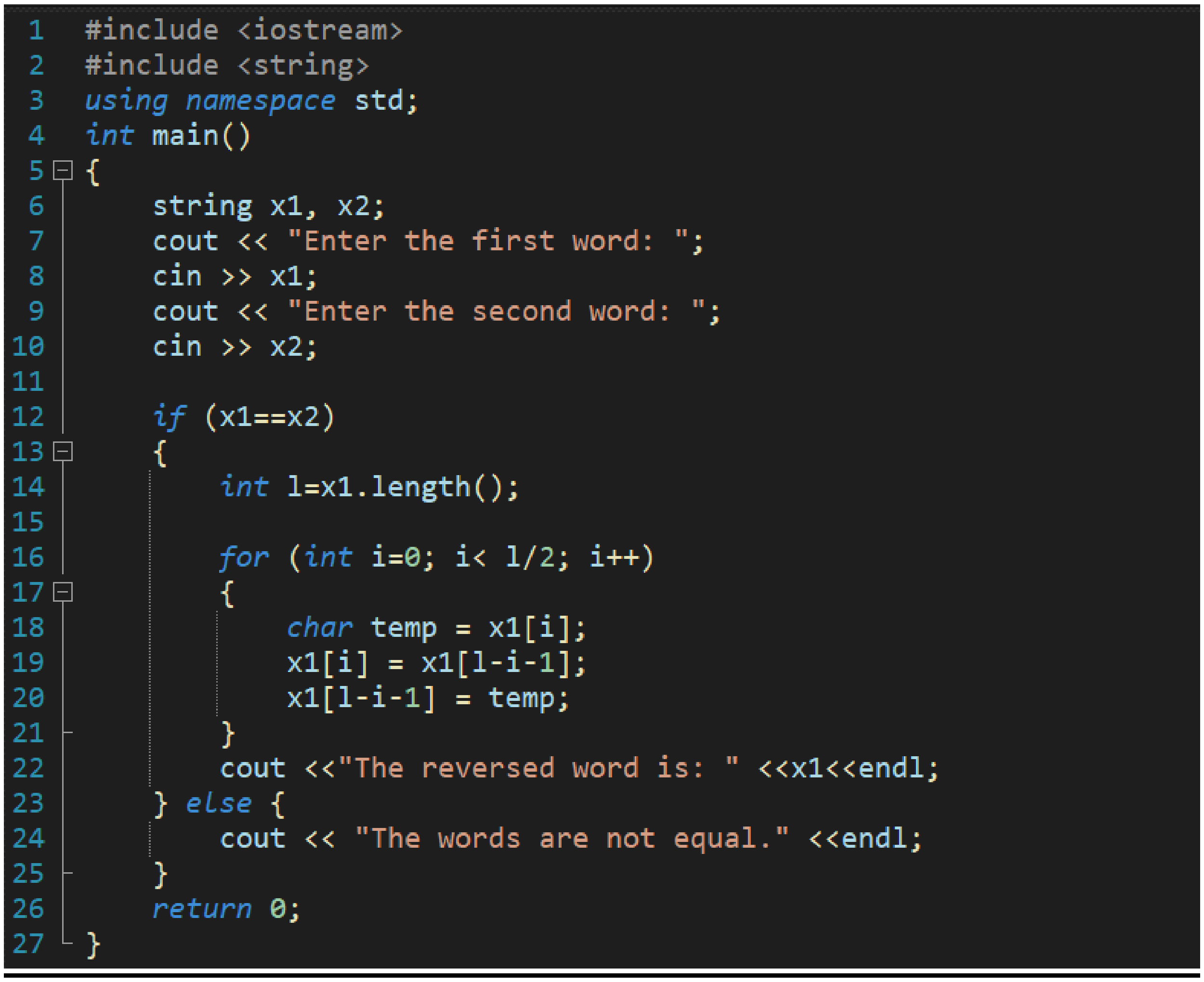
Course Title: Fundamentals of Programming
Course code: CS-109
Submitted to: Ms. Laiba Waheed

Department of Aerospace Engineering
School of Mechanical and Manufacturing Engineering
National University of Science and Technology
H12, Islamabad

Contents

<u>Question 1:</u>

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.

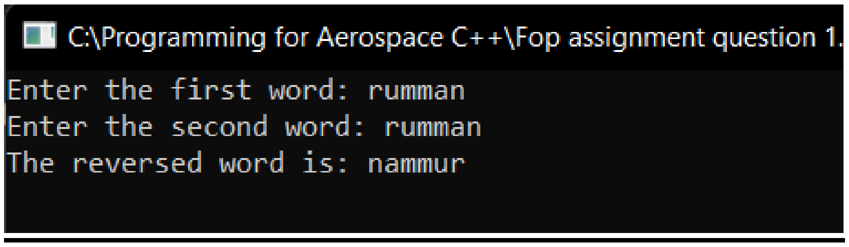


```cpp
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string x1, x2;
    cout << "Enter the first word: ";
    cin >> x1;
    cout << "Enter the second word: ";
    cin >> x2;

    if (x1==x2)
    {
        int l=x1.length();

        for (int i=0; i< l/2; i++)
        {
            char temp = x1[i];
            x1[i] = x1[l-i-1];
            x1[l-i-1] = temp;
        }
        cout <<"The reversed word is: " <<x1<<endl;
    } else {
        cout << "The words are not equal." <<endl;
    }
    return 0;
}
```

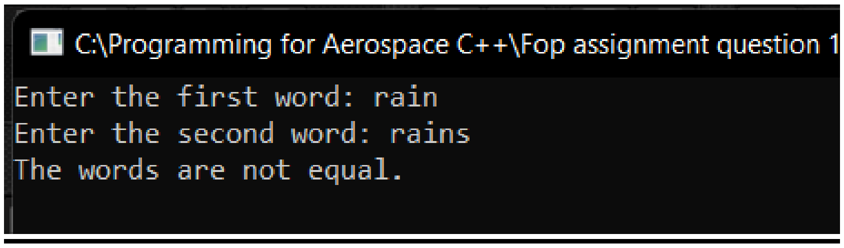
Code explanation:

- The code declares two string variables x1 and x2 and asks the user to input two words and stores the words in these variables.
- It then checks if the two input words (x1 and x2) are equal.
- If they are equal, it calculates the length of the first word (x1.length()).
- It then performs a reversal of the first word (x1) by swapping characters from the beginning to the middle of the word with characters from the end to the middle of the word.
- The output of the reversed word is displayed. If the words are not equal, it the code notifies the user that the words are different

Output:



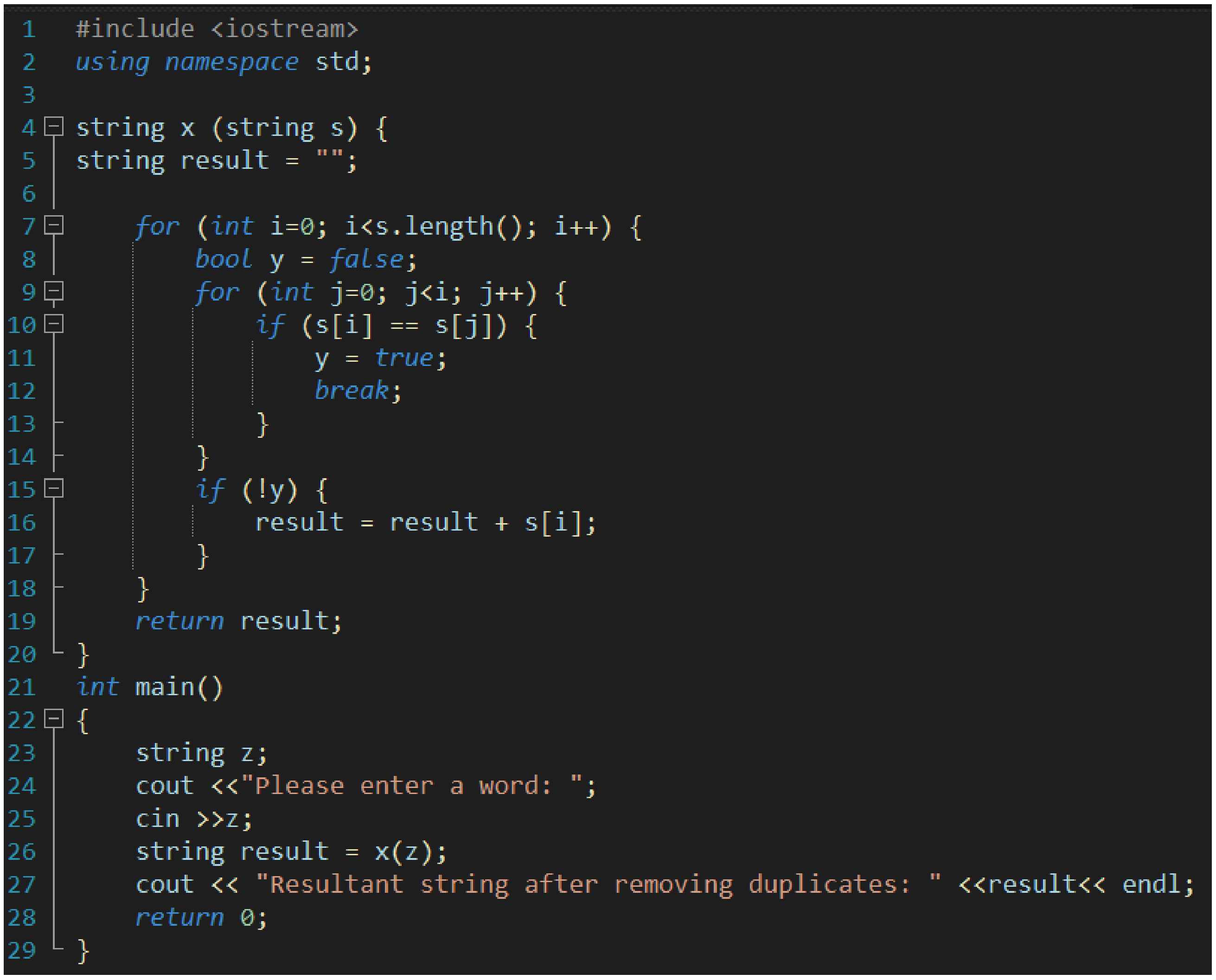
```
C:\Programming for Aerospace C++\Fop assignment question 1.
Enter the first word: rumman
Enter the second word: rumman
The reversed word is: nammur
```



```
C:\Programming for Aerospace C++\Fop assignment question 1
Enter the first word: rain
Enter the second word: rains
The words are not equal.
```

Question 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.



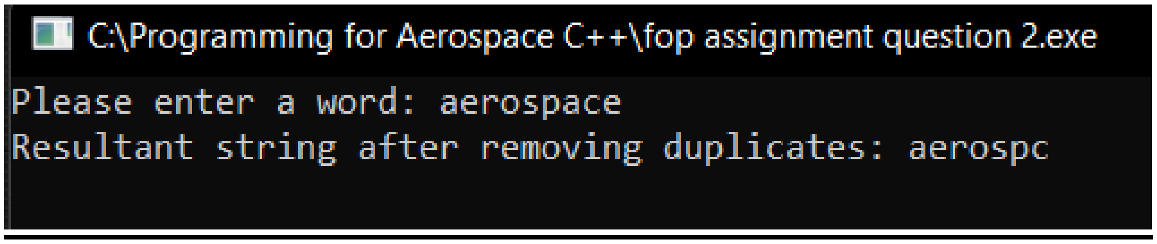
```cpp
#include <iostream>
using namespace std;

string x (string s) {
string result = "";

    for (int i=0; i<s.length(); i++) {
        bool y = false;
        for (int j=0; j<i; j++) {
            if (s[i] == s[j]) {
                y = true;
                break;
            }
        }
        if (!y) {
            result = result + s[i];
        }
    }
    return result;
}
int main()
{
    string z;
    cout <<"Please enter a word: ";
    cin >>z;
    string result = x(z);
    cout << "Resultant string after removing duplicates: " <<result<< endl;
    return 0;
}
```

Code explanation:

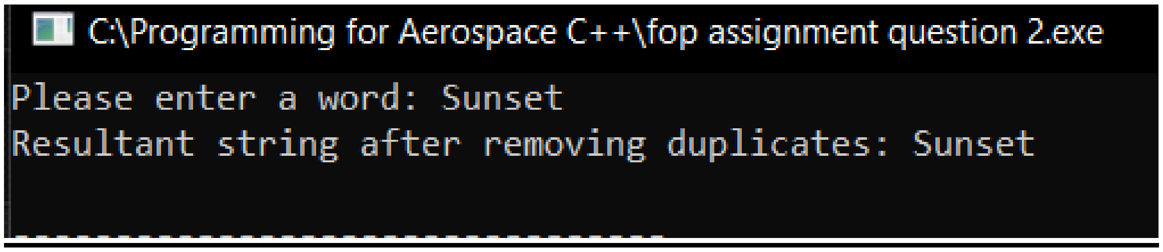
- The code defines a function x that takes a string s as input and returns a string "result" without any duplicates
- The "result" string is created to store the new modified string that doesn't contain any duplicates
- Next, a nested loop is used to iterate through the characters of the inputted word, s.
- The outer loop iterates each character of the input word, and the inner loop is used to check for duplicates within the string before the current index i.

- If a character at index i matches a character before the index i, the boolean variable y is set to true which indicates that there is a duplicate.
- If no match is found, the boolean is false and the character is stored in the "result" string.
- In the main function, the user is asked to enter a word. The function x, as defined above, is called with the inputted user word and the outcome is stored in the "result" variable containing no duplicates.
- The output displays the user inputted word with no characters repeating.

Output:



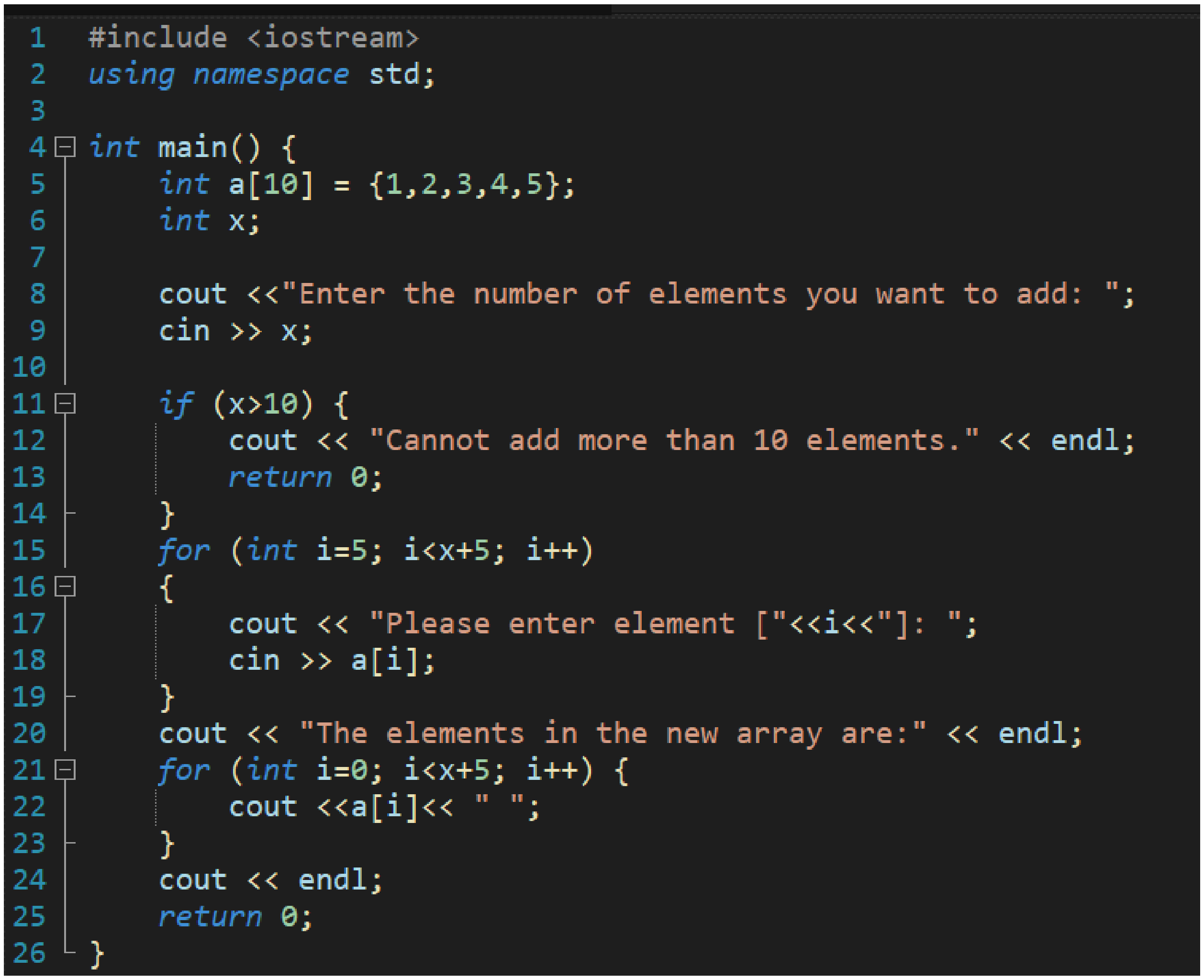
```
C:\Programming for Aerospace C++\fop assignment question 2.exe
Please enter a word: aerospace
Resultant string after removing duplicates: aerospc
```



```
C:\Programming for Aerospace C++\fop assignment question 2.exe
Please enter a word: Sunset
Resultant string after removing duplicates: Sunset
```

Question 3:

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



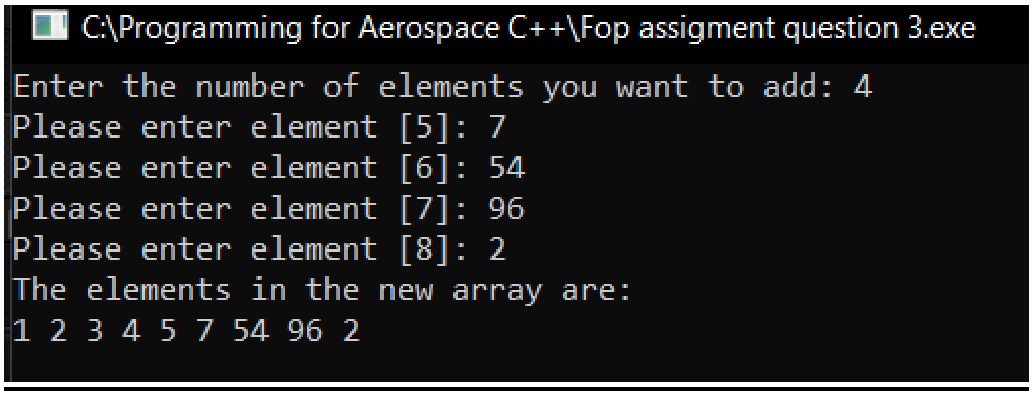
```cpp
#include <iostream>
using namespace std;

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

    cout <<"Enter the number of elements you want to add: ";
    cin >> x;

    if (x>10) {
        cout << "Cannot add more than 10 elements." << endl;
        return 0;
    }
    for (int i=5; i<x+5; i++)
    {
        cout << "Please enter element ["<<i<<"]: ";
        cin >> a[i];
    }
    cout << "The elements in the new array are:" << endl;
    for (int i=0; i<x+5; i++) {
        cout <<a[i]<< " ";
    }
    cout << endl;
    return 0;
}
```

Code explanation:

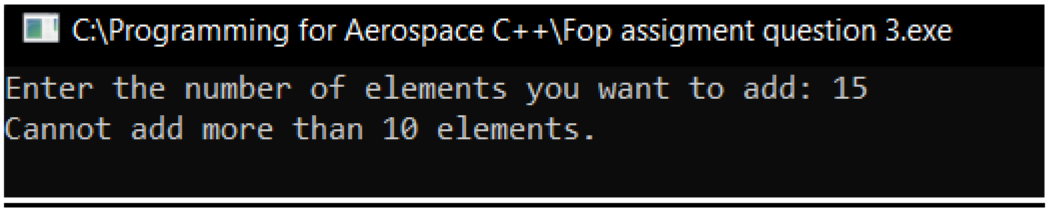
- The array int a[15] = {1,2,3,4,5} is initialized with a size of 15 and only the first 5 elements with values {1, 2, 3, 4, 5} are initialized.
- The user is asked to declare an integer variable x, indicating the number of elements they want to add to the array. If the user wants to enter more than 10 elements, the program sends an error message. If they want to enter less than 10 elements, the code proceeds
- A for loop is then used, starting from index 5 because the first 5 elements of the array a are already initialized. It continues up to x + 5 (the number of elements the user wants to add plus the starting index).
- Inside the loop, the user is asked to input elements for positions after the initial given array elements.

- Finally, a message indicating the elements in the new array will he user added elements will be displayed.

Output:



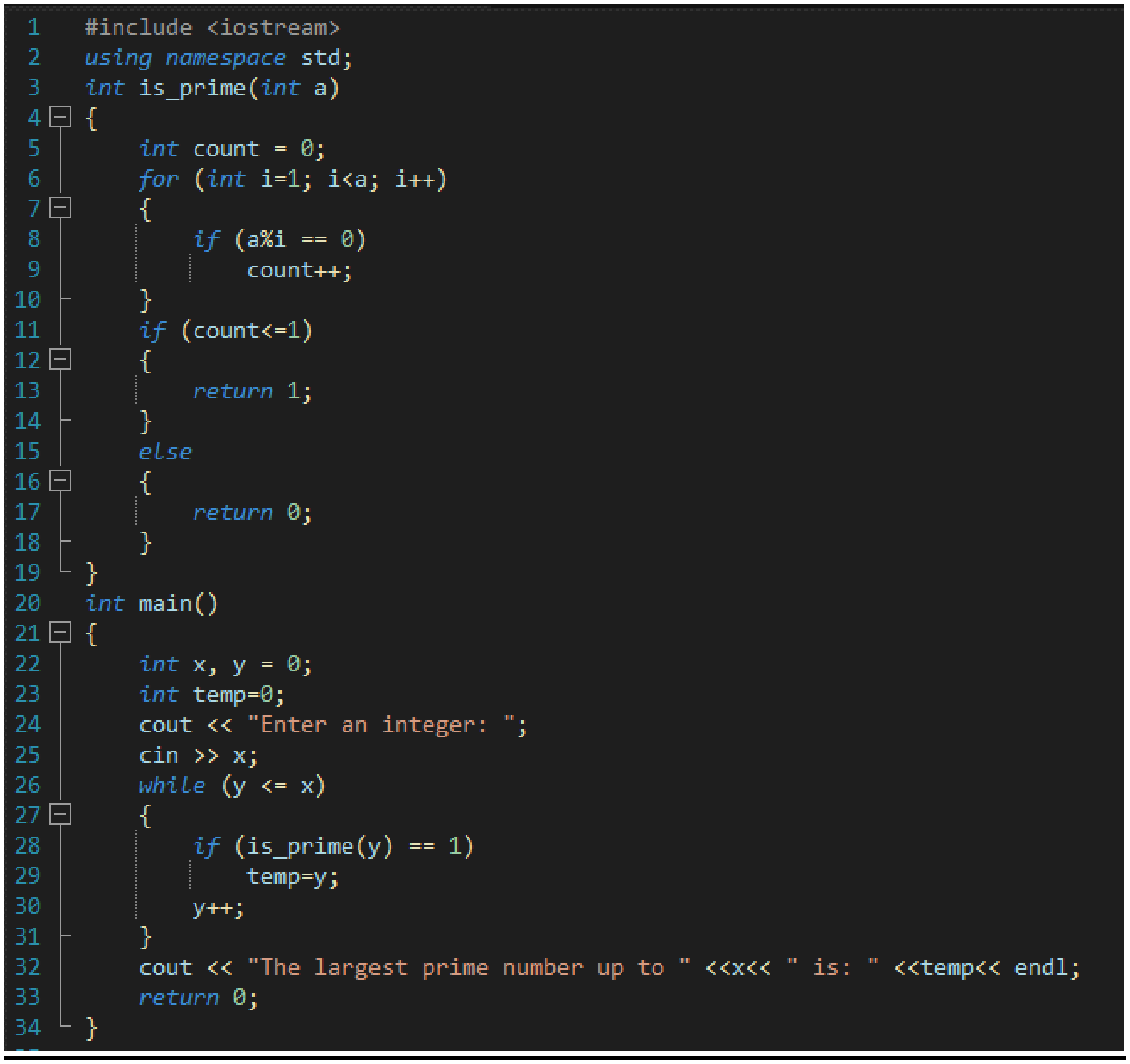
```
C:\Programming for Aerospace C++\Fop assigment question 3.exe
Enter the number of elements you want to add: 4
Please enter element [5]: 7
Please enter element [6]: 54
Please enter element [7]: 96
Please enter element [8]: 2
The elements in the new array are:
1 2 3 4 5 7 54 96 2
```



```
C:\Programming for Aerospace C++\Fop assigment question 3.exe
Enter the number of elements you want to add: 15
Cannot add more than 10 elements.
```

Question 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.



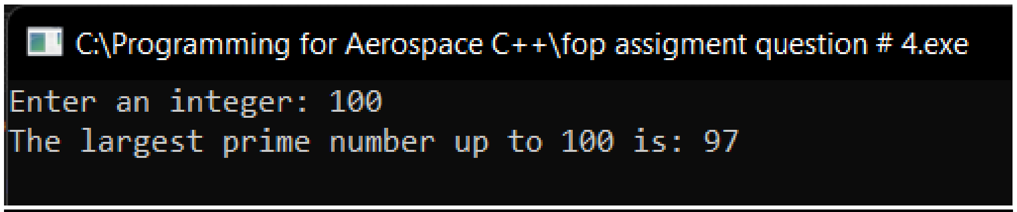
```cpp
#include <iostream>
using namespace std;
int is_prime(int a)
{
    int count = 0;
    for (int i=1; i<a; i++)
    {
        if (a%i == 0)
            count++;
    }
    if (count<=1)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
int main()
{
    int x, y = 0;
    int temp=0;
    cout << "Enter an integer: ";
    cin >> x;
    while (y <= x)
    {
        if (is_prime(y) == 1)
            temp=y;
        y++;
    }
    cout << "The largest prime number up to " <<x<< " is: " <<temp<< endl;
    return 0;
}
```

Code explanation:

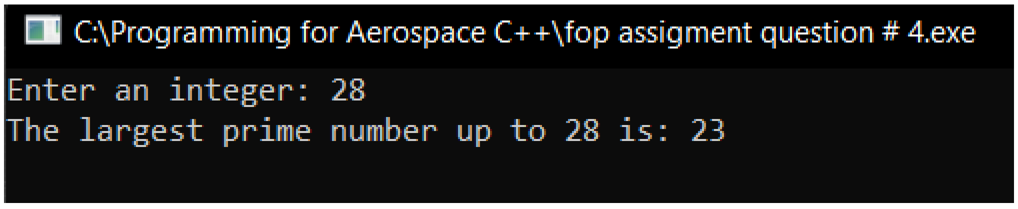
- A function is used to take an integer a to check if it's a prime number or not. It initializes a counter variable count to 0 and iterates through numbers from 1 to a-1 using a loop
- For each i in the loop, it checks if a is divisible by i without leaving a remainder (if (a % i == 0)). If so, it increments the count variable.
- After the loop, if count is <= 1 (meaning a is divisible by exactly 1 or itself), it returns 1, indicating that the number is prime. Otherwise, it returns 0.

- In the main function, variables x, y, and temp are declared, and y and temp are initialized to 0. The user is asked to input the value of the variable x.
- A while loop is then run as long as y <= x. Inside the loop, it is checked if the value of y is a prime number by calling the is_prime function made earlier. If the number is prime, the temp variable is updated with the current value of y. y is then incremented after each loop iteration.
- The value of temp is displayed as the result, indicating the largest prime number up to the user entered integer.

Output:



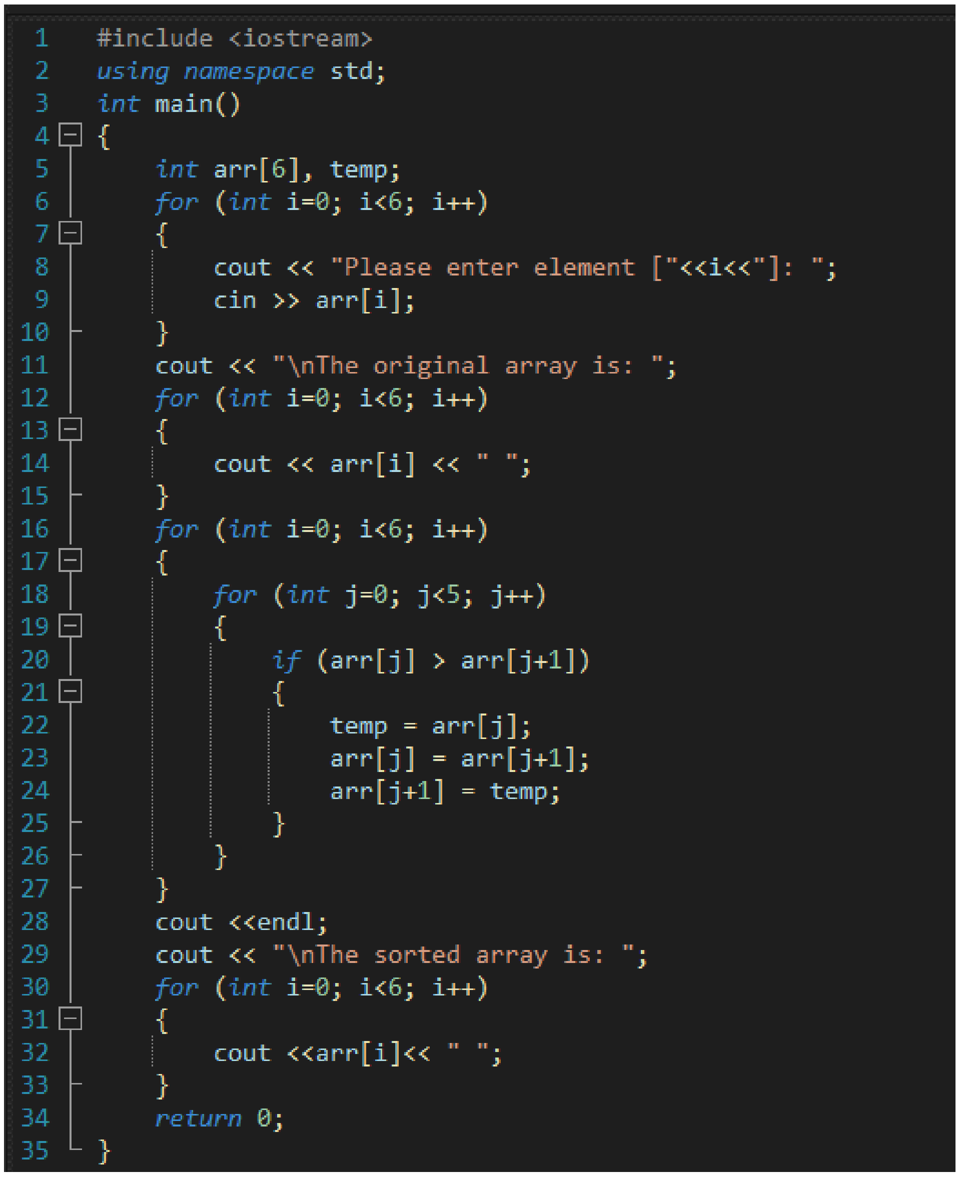
```
C:\Programming for Aerospace C++\fop assigment question # 4.exe
Enter an integer: 100
The largest prime number up to 100 is: 97
```



```
C:\Programming for Aerospace C++\fop assigment question # 4.exe
Enter an integer: 28
The largest prime number up to 28 is: 23
```

Question 5:

Implement Bubble Sort on an array of 6 integers.



```cpp
#include <iostream>
using namespace std;
int main()
{
    int arr[6], temp;
    for (int i=0; i<6; i++)
    {
        cout << "Please enter element ["<<i<<"]: ";
        cin >> arr[i];
    }
    cout << "\nThe original array is: ";
    for (int i=0; i<6; i++)
    {
        cout << arr[i] << " ";
    }
    for (int i=0; i<6; i++)
    {
        for (int j=0; j<5; j++)
        {
            if (arr[j] > arr[j+1])
            {
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    cout <<endl;
    cout << "\nThe sorted array is: ";
    for (int i=0; i<6; i++)
    {
        cout <<arr[i]<< " ";
    }
    return 0;
}
```

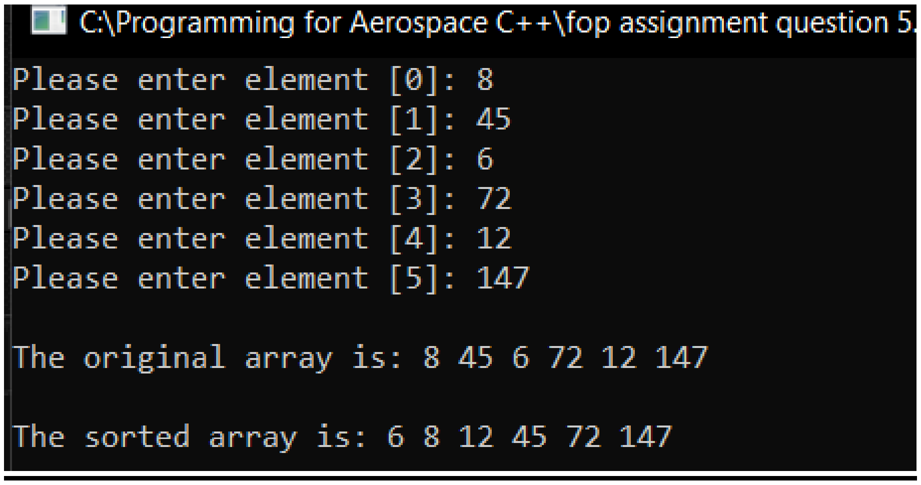
Code explanation:

- An array of size 6 is declared, and a for loop is used to ask the user to input elements into the array.
- Then, a bubble sort is used. A nested for loop compares adjacent elements to each other and swaps them in ascending order.
- The outer loop is used to go through each element of the array and the inner loop is used for comparison between adjacent elements. It goes up to 5 because in each iteration of the

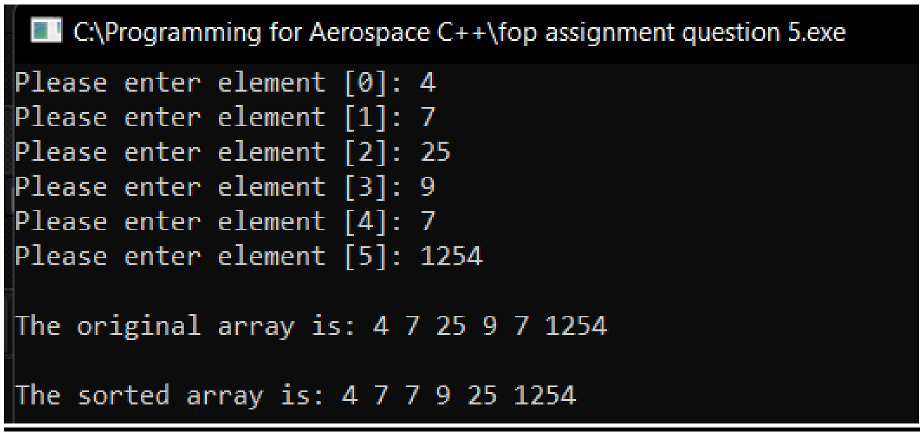
outer loop, the largest element gets placed at the end, so the last element is already in its correct sorted position.

- The elements of the array in ascending order are displayed on the screen as output.

Output:



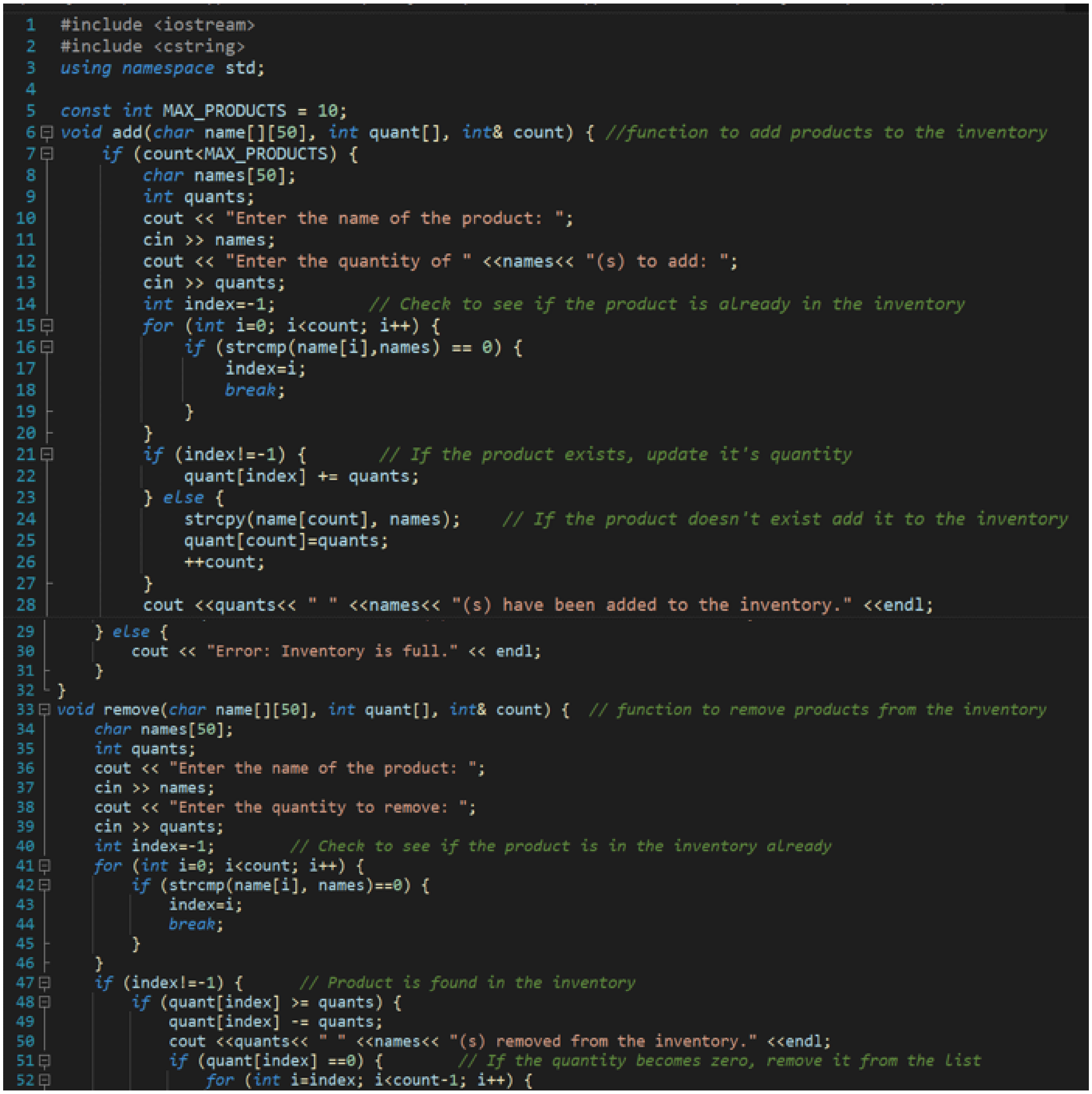
```
C:\Programming for Aerospace C++\fop assignment question 5.
Please enter element [0]: 8
Please enter element [1]: 45
Please enter element [2]: 6
Please enter element [3]: 72
Please enter element [4]: 12
Please enter element [5]: 147

The original array is: 8 45 6 72 12 147

The sorted array is: 6 8 12 45 72 147
```



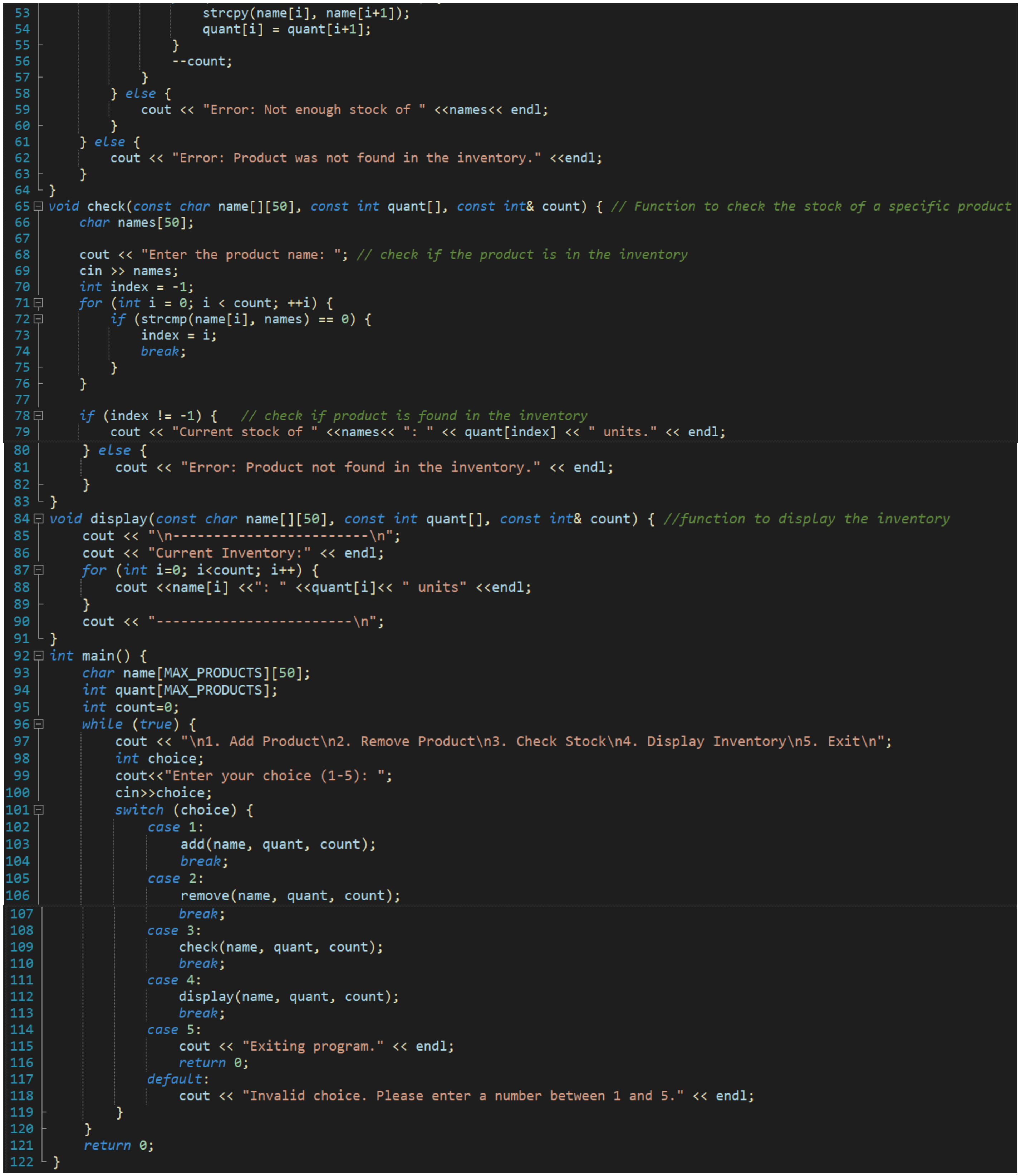
```
C:\Programming for Aerospace C++\fop assignment question 5.exe
Please enter element [0]: 4
Please enter element [1]: 7
Please enter element [2]: 25
Please enter element [3]: 9
Please enter element [4]: 7
Please enter element [5]: 1254

The original array is: 4 7 25 9 7 1254

The sorted array is: 4 7 7 9 25 1254
```

Question 6:

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



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

const int MAX_PRODUCTS = 10;
void add(char name[][50], int quant[], int& count) { //function to add products to the inventory
    if (count<MAX_PRODUCTS) {
        char names[50];
        int quants;
        cout << "Enter the name of the product: ";
        cin >> names;
        cout << "Enter the quantity of " <<names<< "(s) to add: ";
        cin >> quants;
        int index=-1;         // Check to see if the product is already in the inventory
        for (int i=0; i<count; i++) {
            if (strcmp(name[i],names) == 0) {
                index=i;
                break;
            }
        }
        if (index!=-1) {        // If the product exists, update it's quantity
            quant[index] += quants;
        } else {
            strcpy(name[count], names);    // If the product doesn't exist add it to the inventory
            quant[count]=quants;
            ++count;
        }
        cout <<quants<< " " <<names<< "(s) have been added to the inventory." <<endl;
    } else {
        cout << "Error: Inventory is full." << endl;
    }
}
void remove(char name[][50], int quant[], int& count) {  // function to remove products from the inventory
    char names[50];
    int quants;
    cout << "Enter the name of the product: ";
    cin >> names;
    cout << "Enter the quantity to remove: ";
    cin >> quants;
    int index=-1;        // Check to see if the product is in the inventory already
    for (int i=0; i<count; i++) {
        if (strcmp(name[i], names)==0) {
            index=i;
            break;
        }
    }
    if (index!=-1) {      // Product is found in the inventory
        if (quant[index] >= quants) {
            quant[index] -= quants;
            cout <<quants<< " " <<names<< "(s) removed from the inventory." <<endl;
            if (quant[index] ==0) {         // If the quantity becomes zero, remove it from the list
                for (int i=index; i<count-1; i++) {
```

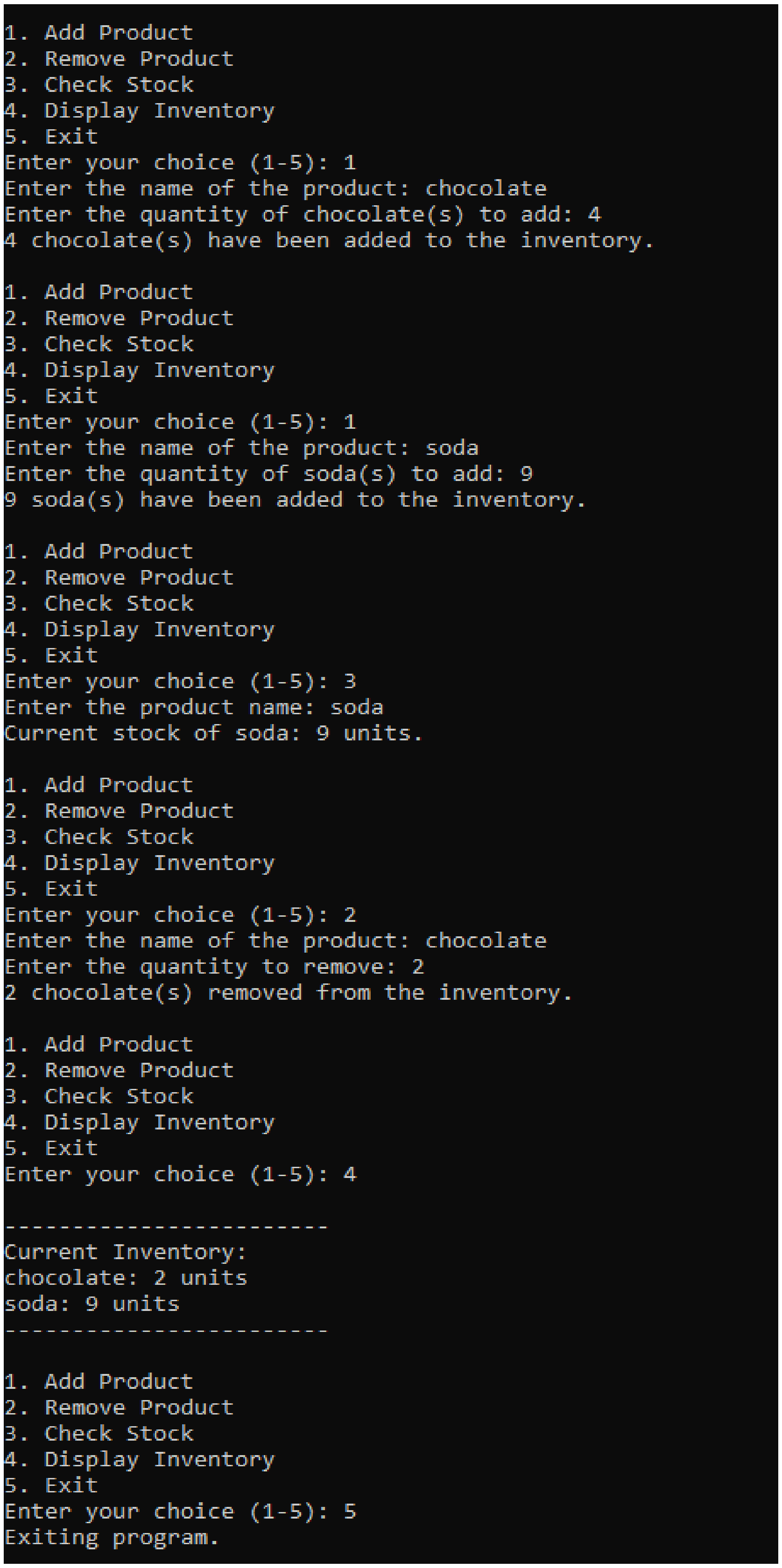


```cpp
                    strcpy(name[i], name[i+1]);
                    quant[i] = quant[i+1];
                }
                --count;
            }
        } else {
            cout << "Error: Not enough stock of " <<names<< endl;
        }
    } else {
        cout << "Error: Product was not found in the inventory." <<endl;
    }
}
void check(const char name[][50], const int quant[], const int& count) { // Function to check the stock of a specific product
    char names[50];

    cout << "Enter the product name: "; // check if the product is in the inventory
    cin >> names;
    int index = -1;
    for (int i = 0; i < count; ++i) {
        if (strcmp(name[i], names) == 0) {
            index = i;
            break;
        }
    }

    if (index != -1) {   // check if product is found in the inventory
        cout << "Current stock of " <<names<< ": " << quant[index] << " units." << endl;
    } else {
        cout << "Error: Product not found in the inventory." << endl;
    }
}
void display(const char name[][50], const int quant[], const int& count) { //function to display the inventory
    cout << "\n------------------------\n";
    cout << "Current Inventory:" << endl;
    for (int i=0; i<count; i++) {
        cout <<name[i] <<": " <<quant[i]<< " units" <<endl;
    }
    cout << "------------------------\n";
}
int main() {
    char name[MAX_PRODUCTS][50];
    int quant[MAX_PRODUCTS];
    int count=0;
    while (true) {
        cout << "\n1. Add Product\n2. Remove Product\n3. Check Stock\n4. Display Inventory\n5. Exit\n";
        int choice;
        cout<<"Enter your choice (1-5): ";
        cin>>choice;
        switch (choice) {
            case 1:
                add(name, quant, count);
                break;
            case 2:
                remove(name, quant, count);
                break;
            case 3:
                check(name, quant, count);
                break;
            case 4:
                display(name, quant, count);
                break;
            case 5:
                cout << "Exiting program." << endl;
                return 0;
            default:
                cout << "Invalid choice. Please enter a number between 1 and 5." << endl;
        }
    }
    return 0;
}
```

Code explanation:

- This code is an inventory management system that allows users to perform various operations on a product inventory. Users could add products, remove products, check the stock of a specific product, and display the current inventory status.
- First, a function is created to allow users to add products to the inventory by specifying the name and quantity. If the product exists in the inventory already, the quantity is updated. If it doesn't exist, the quantity is added to the inventory.
- Next, a function is created to allow users to remove products to the inventory. If the product exists in the inventory already, the quantity is reduced. If the quantity becomes zero, it removes the product from the list.
- The check function is then created to check the stock of a particular product.
- The display function is created to show the current status of the entire inventory, showing the names and the quantities of the products in the inventory.
- The main function contains a while loop that continuously asks a user to input a choice from 1 to 5 to perform the corresponding operation of adding, removing, checking, or displaying products in the inventory. The 5th option is to exit the program.
- Once the code is executed, users are presented with the option to pick whatever option they wish to perform for the inventory until they chose to exit it.

Output:



```
1. Add Product
2. Remove Product
3. Check Stock
4. Display Inventory
5. Exit
Enter your choice (1-5): 1
Enter the name of the product: chocolate
Enter the quantity of chocolate(s) to add: 4
4 chocolate(s) have been added to the inventory.

1. Add Product
2. Remove Product
3. Check Stock
4. Display Inventory
5. Exit
Enter your choice (1-5): 1
Enter the name of the product: soda
Enter the quantity of soda(s) to add: 9
9 soda(s) have been added to the inventory.

1. Add Product
2. Remove Product
3. Check Stock
4. Display Inventory
5. Exit
Enter your choice (1-5): 3
Enter the product name: soda
Current stock of soda: 9 units.

1. Add Product
2. Remove Product
3. Check Stock
4. Display Inventory
5. Exit
Enter your choice (1-5): 2
Enter the name of the product: chocolate
Enter the quantity to remove: 2
2 chocolate(s) removed from the inventory.

1. Add Product
2. Remove Product
3. Check Stock
4. Display Inventory
5. Exit
Enter your choice (1-5): 4

------------------------
Current Inventory:
chocolate: 2 units
soda: 9 units
------------------------

1. Add Product
2. Remove Product
3. Check Stock
4. Display Inventory
5. Exit
Enter your choice (1-5): 5
Exiting program.
```