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 (CS109)

Lab Report: #5

“Arrays (Part-01)”

Name: ***Zoha Maryam***

CMS: **455672**

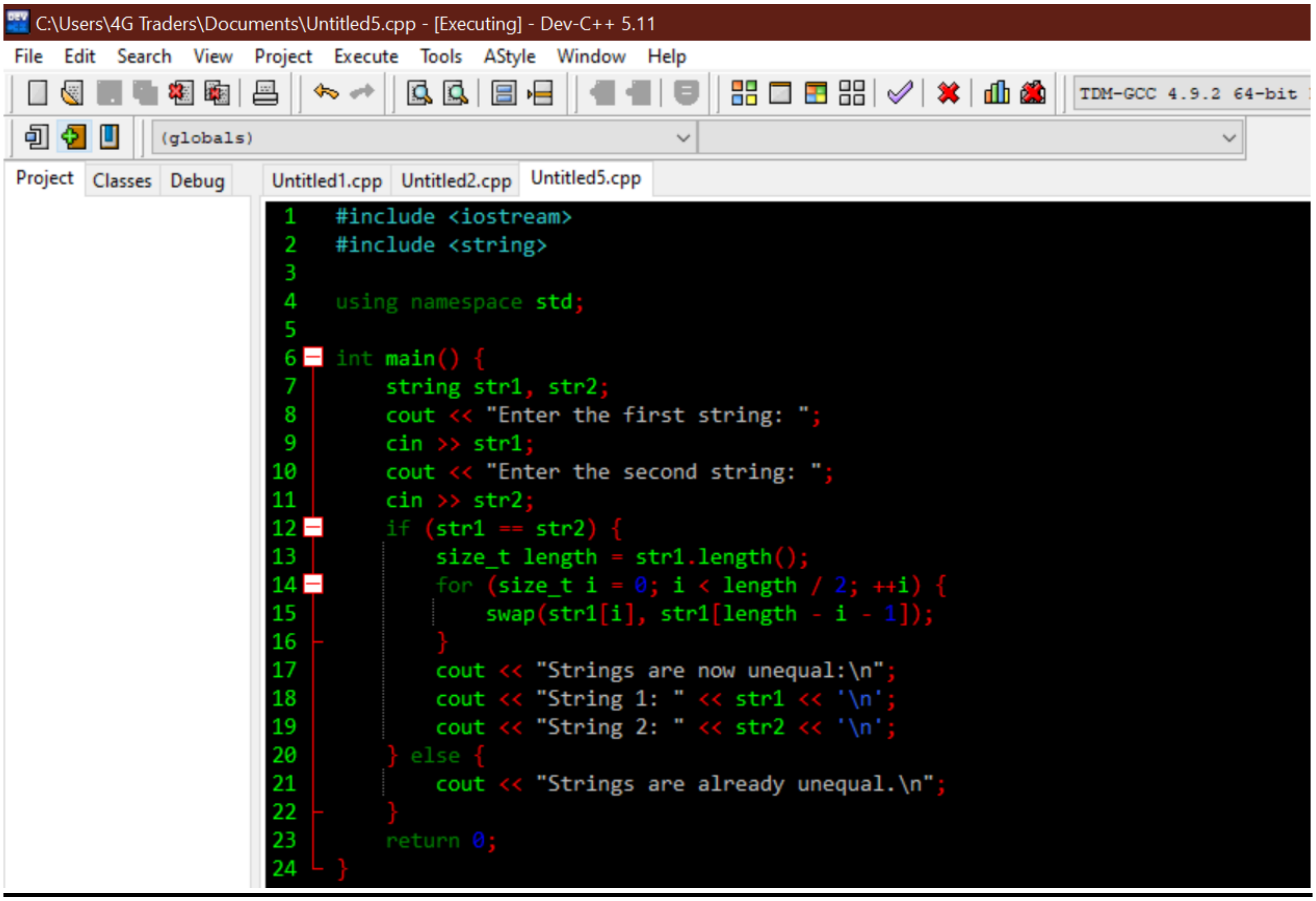
Q2 ._Write a C++ program, take two strings as input from user and check if both strings are equal or not. If they are equal make them unequal by rotating string. e.g., Hello is turned into olleH etc.

Introduction:

This C++ program takes two strings as input from the user, checks if they are equal, and if so, reverses the first string to make them unequal. The reversal is done manually without using the swap function or any other standard library functions.

Program:

```
#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) {
        size_t length = str1.length();
        for (size_t i = 0; i < length / 2; ++i)
 {
            swap(str1[i], str1[length - i - 1]);
        }
        cout << "Strings are now unequal:\n";
        cout << "String 1: " << str1 << '\n';
        cout << "String 2: " << str2 << '\n';
    }
 else
 {
        cout << "Strings are already unequal.\n";
    }
    return 0;
}
```

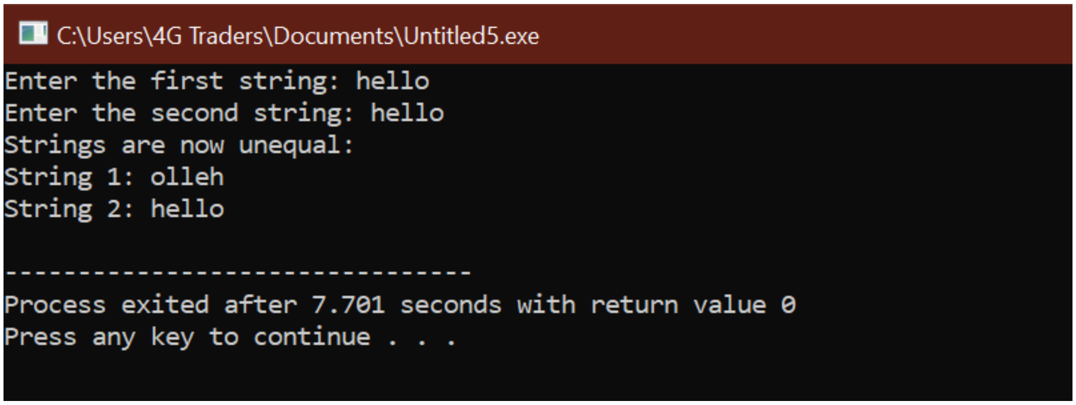


```cpp
#include <iostream>
#include <string>

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) {
        size_t length = str1.length();
        for (size_t i = 0; i < length / 2; ++i) {
            swap(str1[i], str1[length - i - 1]);
        }
        cout << "Strings are now unequal:\n";
        cout << "String 1: " << str1 << '\n';
        cout << "String 2: " << str2 << '\n';
    } else {
        cout << "Strings are already unequal.\n";
    }
    return 0;
}
```

Logic Explanation:

- The program prompts the user to enter two strings, str1 and str2.
- It checks if the input strings are equal.
- If the strings are equal, it manually reverses the first string by swapping characters from the beginning to the middle.
- The reversed string and the original second string are then displayed.
- If the strings are already unequal, a message is displayed indicating that they are already unequal

Output:



```
Enter the first string: hello
Enter the second string: hello
Strings are now unequal:
String 1: olleh
String 2: hello

--------------------------------
Process exited after 7.701 seconds with return value 0
Press any key to continue . . .
```

Output Explanation:

If you enter "hello" for both strings, the output will be:

```
 Strings are now unequal:
String 1: olleh
String 2: hello
```

This indicates that the program reversed the first string to make them unequal.

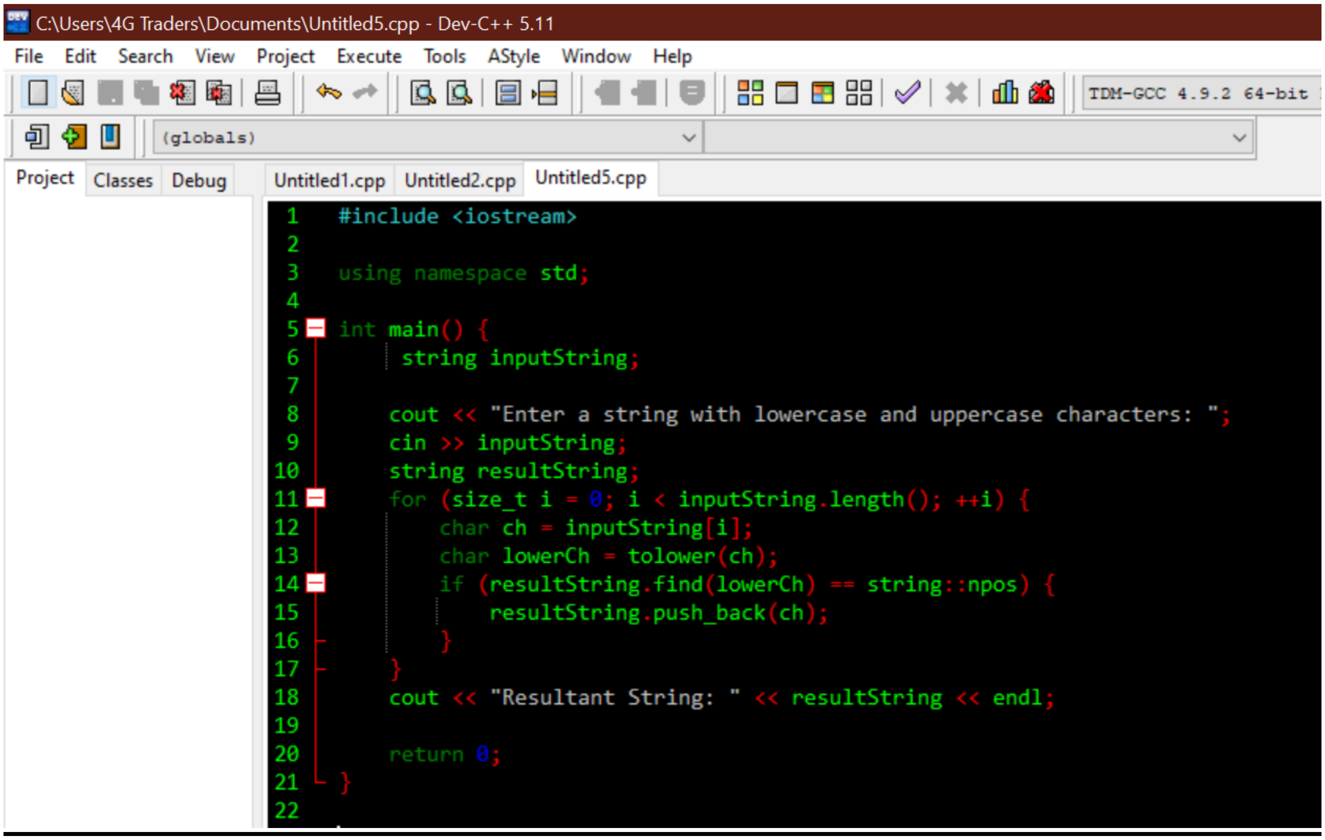
Q2 .Write a C++program for a string which may contain lowercase and uppercase characters. The task is to remove all duplicate characters from the string and find the resultant string.

Introduction:

This C++ program takes a string as input, which may contain both lowercase and uppercase characters. The task is to remove all duplicate characters from the string and find the resultant string. The program uses a basic loop to iterate through each character in the input string, checks for duplicates, and constructs the result string.

Program:

```
#include <iostream>

using namespace std;

int main() {
     string inputString;

    cout << "Enter a string with lowercase and uppercase characters: ";
    cin >> inputString;
    string resultString;
    for (size_t i = 0; i < inputString.length(); ++i) {
        char ch = inputString[i];
        char lowerCh = tolower(ch);
        if (resultString.find(lowerCh) == string::npos) {
            resultString.push_back(ch);
        }
    }
    cout << "Resultant String: " << resultString << endl;

    return 0;
}
```

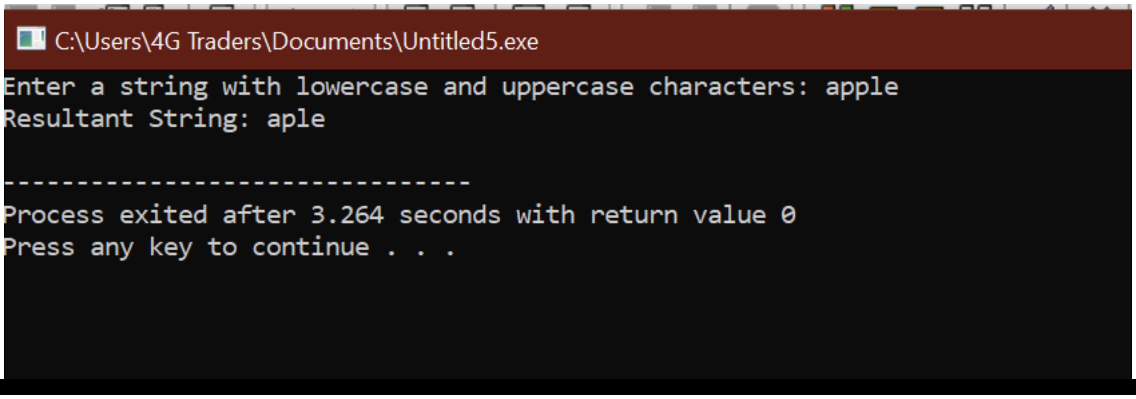


```cpp
#include <iostream>

using namespace std;

int main() {
     string inputString;

    cout << "Enter a string with lowercase and uppercase characters: ";
    cin >> inputString;
    string resultString;
    for (size_t i = 0; i < inputString.length(); ++i) {
        char ch = inputString[i];
        char lowerCh = tolower(ch);
        if (resultString.find(lowerCh) == string::npos) {
            resultString.push_back(ch);
        }
    }
    cout << "Resultant String: " << resultString << endl;

    return 0;
}
```

Logic Explanation:

- The program uses cin to get a string input from the user.
- It initializes an empty string resultString to store the characters without duplicates.
- It iterates through each character in the input string using an index-based loop.
- For each character, it converts it to lowercase using tolower for case-insensitive comparison.
- It checks if the lowercase character is encountered for the first time in the resultString.
- If the character is not already present, it adds the character to the resultString.
- The final resultString contains unique characters from the input string.
- The program outputs the resultant string without duplicate characters.

Output:



```
Enter a string with lowercase and uppercase characters: apple
Resultant String: aple

--------------------------------
Process exited after 3.264 seconds with return value 0
Press any key to continue . . .
```

Output Explanation:

If you enter the string "apple", the output will be:
Resultant String will be aple.
This indicates that the resultant string contains unique characters from the input string while maintaining the order of their appearance

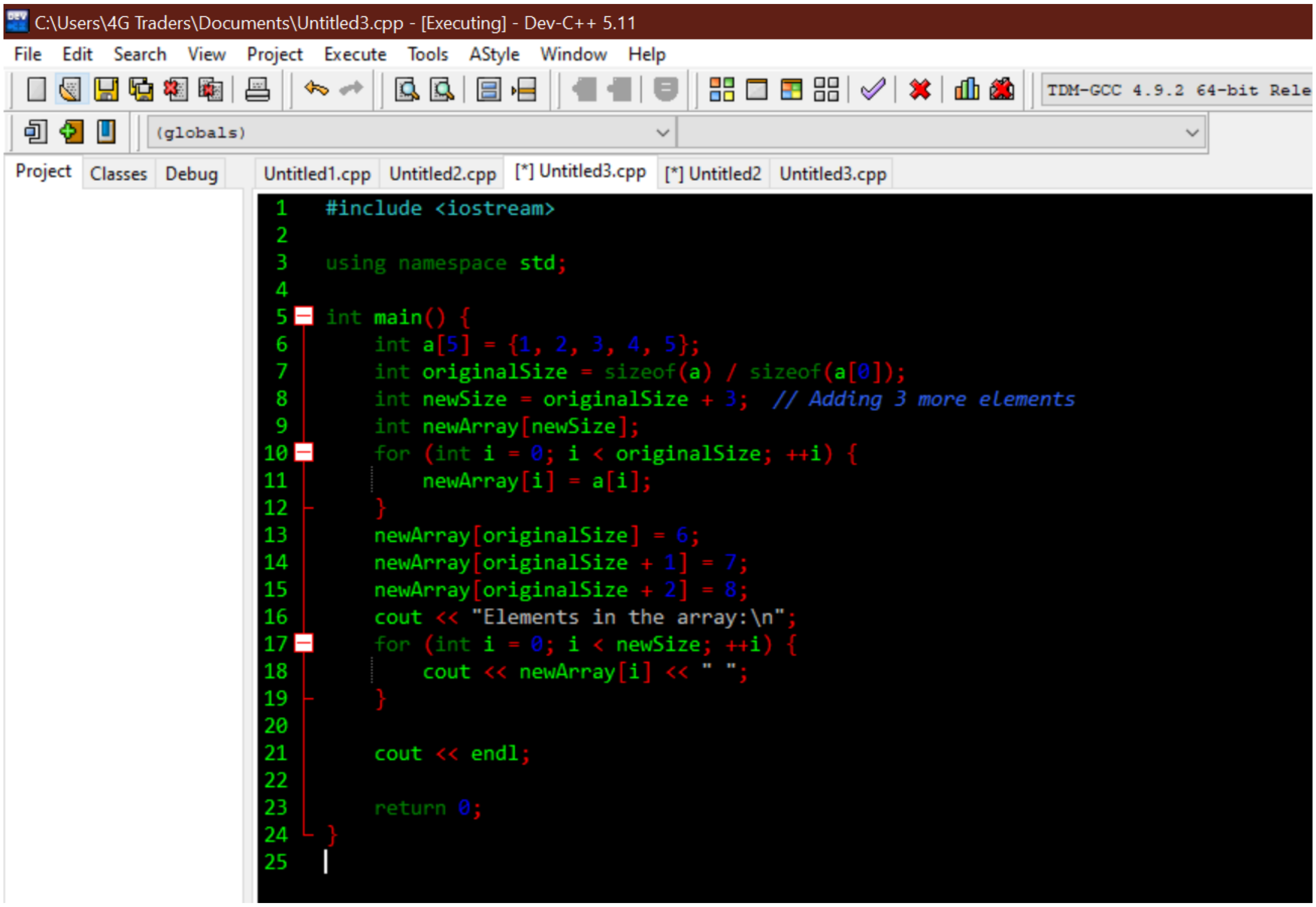
Q4 . Suppose an integer array a[5] = {1,2,3,4,5}. Add more elements to it and display them in C++.

Introduction:

This C++ program initializes an integer array **a** with 5 elements and then adds 3 more elements to it, creating a new array **newArray**. The program displays all the elements in the combined array.

Program:

```
#include <iostream>

using namespace std;

int main() {
    int a[5] = {1, 2, 3, 4, 5};
    int originalSize = sizeof(a) / sizeof(a[0]);
    int newSize = originalSize + 3;  // Adding 3 more elements
    int newArray[newSize];
    for (int i = 0; i < originalSize; ++i) {
        newArray[i] = a[i];
    }
    newArray[originalSize] = 6;
    newArray[originalSize + 1] = 7;
    newArray[originalSize + 2] = 8;
    cout << "Elements in the array:\n";
    for (int i = 0; i < newSize; ++i) {
        cout << newArray[i] << " ";
    }

    cout << endl;

    return 0;
}
```

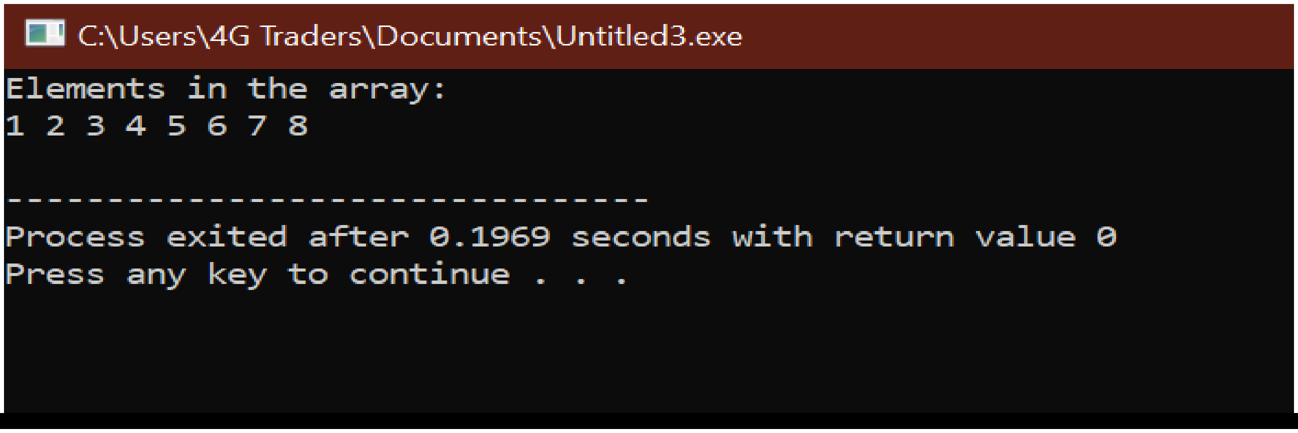

C:\Users\4G Traders\Documents\Untitled3.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Rele
(globals)
Project Classes Debug
Untitled1.cpp Untitled2.cpp [*] Untitled3.cpp [*] Untitled2 Untitled3.cpp

```cpp
#include <iostream>

using namespace std;

int main() {
    int a[5] = {1, 2, 3, 4, 5};
    int originalSize = sizeof(a) / sizeof(a[0]);
    int newSize = originalSize + 3;  // Adding 3 more elements
    int newArray[newSize];
    for (int i = 0; i < originalSize; ++i) {
        newArray[i] = a[i];
    }
    newArray[originalSize] = 6;
    newArray[originalSize + 1] = 7;
    newArray[originalSize + 2] = 8;
    cout << "Elements in the array:\n";
    for (int i = 0; i < newSize; ++i) {
        cout << newArray[i] << " ";
    }

    cout << endl;

    return 0;
}
```

Logic Explanation:

- The program initializes an array a with 5 elements.
- It calculates the size of the new array (newSize) by adding 3 more elements to the original size.
- A new array newArray is created with the larger size.
- Elements from the original array are copied to the new array.
- Additional elements (6, 7, and 8) are added to the new array.
- The program then displays all elements in the combined array.

Output:



```
Elements in the array:
1 2 3 4 5 6 7 8

--------------------------------
Process exited after 0.1969 seconds with return value 0
Press any key to continue . . .
```

Output Explanation:

- The program initializes an array a with 5 elements: {1, 2, 3, 4, 5}. It then adds 3 more elements (6, 7, and 8) to create a new array newArray.
- This output shows the elements present in the combined array. Here's a breakdown of the output:
- The original elements from array a (1, 2, 3, 4, 5) are copied to newArray.
- Additional elements (6, 7, and 8) are added to newArray.
- The cout statements display each element of newArray separated by a space.

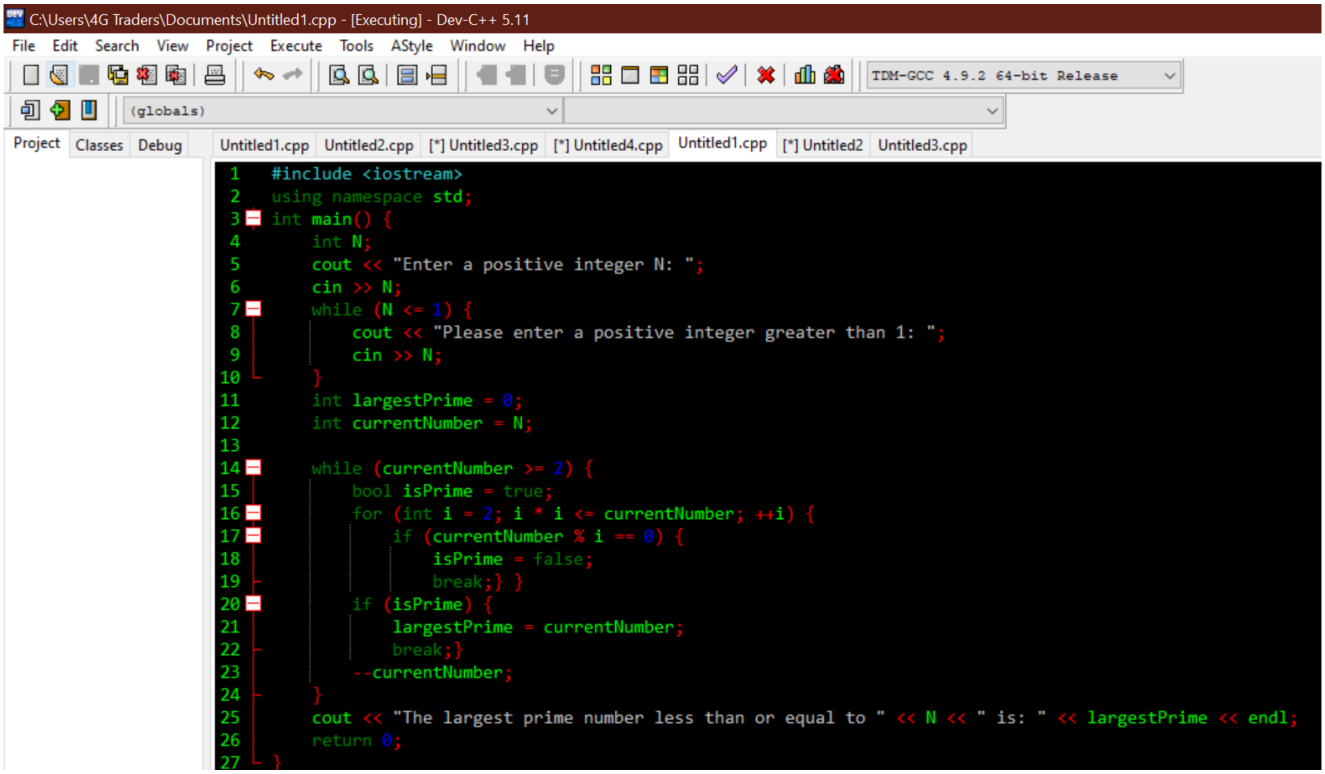
Q4. Write a C++ program that uses a while loop to find the largest prime number less than a given positive integer N. Your program should take the value of N as input from the user and then find the largest prime number less than or equal to N. You are not allowed to use any library or pre-existing functions to check for prime numbers.

Introduction:

This C++ program takes a positive integer N as input from the user and uses a while loop to find the largest prime number less than or equal to N. The program does not use any library or pre-existing functions to check for prime numbers.

Program:

```
#include <iostream>
using namespace std;
int main() {
    int N;
    cout << "Enter a positive integer N: ";
    cin >> N;
    while (N <= 1) {
        cout << "Please enter a positive integer greater than 1: ";
        cin >> N;
    }
    int largestPrime = 0;
    int currentNumber = N;

    while (currentNumber >= 2) {
        bool isPrime = true;
        for (int i = 2; i * i <= currentNumber; ++i) {
            if (currentNumber % i == 0) {
                isPrime = false;
                break;} }
        if (isPrime) {
            largestPrime = currentNumber;
            break;}
        --currentNumber;
    }
    cout << "The largest prime number less than or equal to " << N << " is: " << largestPrime << endl;
    return 0;
}
```

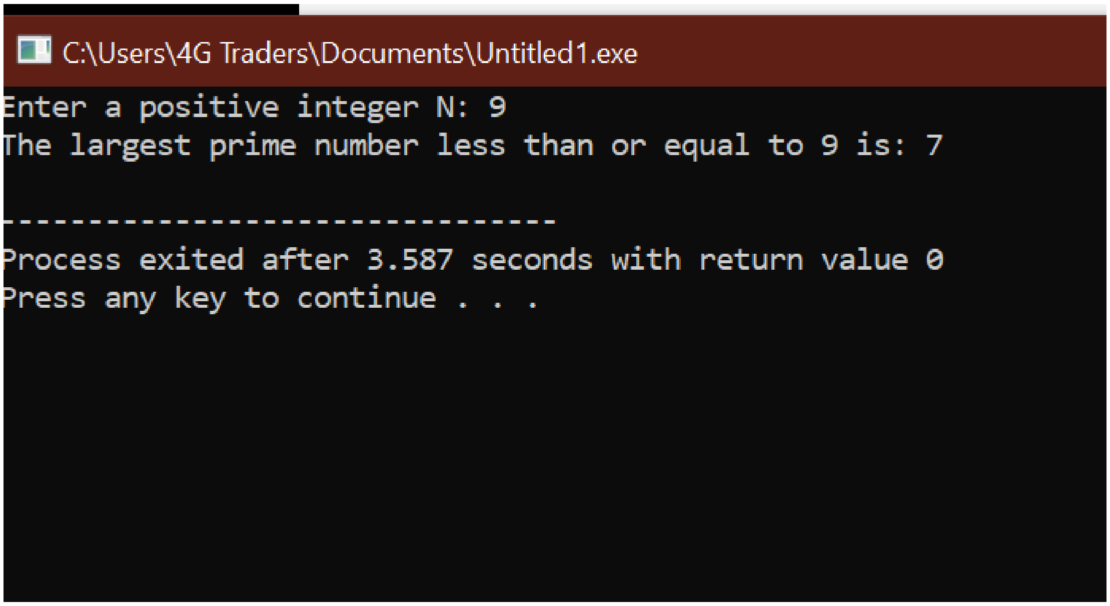

C:\Users\4G Traders\Documents\Untitled1.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Untitled1.cpp Untitled2.cpp [*] Untitled3.cpp [*] Untitled4.cpp Untitled1.cpp [*] Untitled2 Untitled3.cpp

```cpp
#include <iostream>
using namespace std;
int main() {
    int N;
    cout << "Enter a positive integer N: ";
    cin >> N;
    while (N <= 1) {
        cout << "Please enter a positive integer greater than 1: ";
        cin >> N;
    }
    int largestPrime = 0;
    int currentNumber = N;

    while (currentNumber >= 2) {
        bool isPrime = true;
        for (int i = 2; i * i <= currentNumber; ++i) {
            if (currentNumber % i == 0) {
                isPrime = false;
                break;} }
        if (isPrime) {
            largestPrime = currentNumber;
            break;}
        --currentNumber;
    }
    cout << "The largest prime number less than or equal to " << N << " is: " << largestPrime << endl;
    return 0;
}
```

Logic Explanation:

- The program takes a positive integer N as input from the user.
- It validates the input to ensure it's a positive integer greater than 1.
- Using a while loop, it iterates through numbers starting from N and checks if each number is prime.
- If a prime number is found, it is stored as the largest prime, and the loop breaks.
- The program outputs the largest prime number less than or equal to N.

Output:



```
Enter a positive integer N: 9
The largest prime number less than or equal to 9 is: 7

--------------------------------
Process exited after 3.587 seconds with return value 0
Press any key to continue . . .
```

Output Explanation:

- If you enter N as 9, the program will output:
- The largest prime number less than or equal to 9 is: 7
- This indicates that 7 is the largest prime number less than or equal to 10, as 2, 3, 5, and 7 are the prime numbers less than or equal to 9

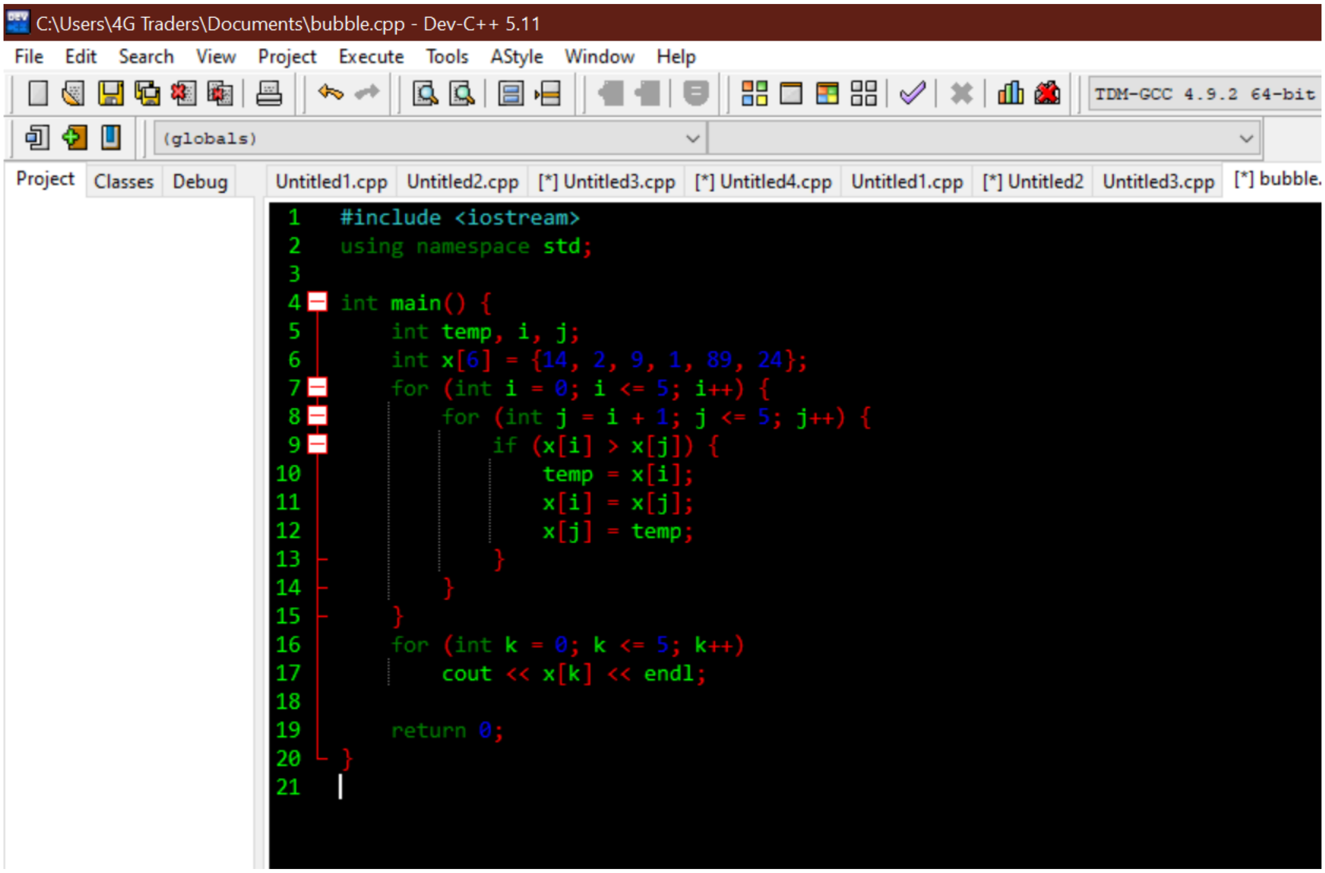
Q5. Implement Bubble Sort on an array of 6 integers. 5

Introduction:

Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. Here, we'll implement Bubble Sort on an array of 6 integers.

Program:

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

int main() {
    int temp, i, j;
    int x[6] = {14, 2, 9, 1, 89, 24};
    for (int i = 0; i <= 5; i++)
{
        for (int j = i + 1; j <= 5; j++)
{
            if (x[i] > x[j]) {
                temp = x[i];
                x[i] = x[j];
                x[j] = temp;
            }
        }
    }
    for (int k = 0; k <= 5; k++)
        cout << x[k] << endl;

    return 0;
}
```

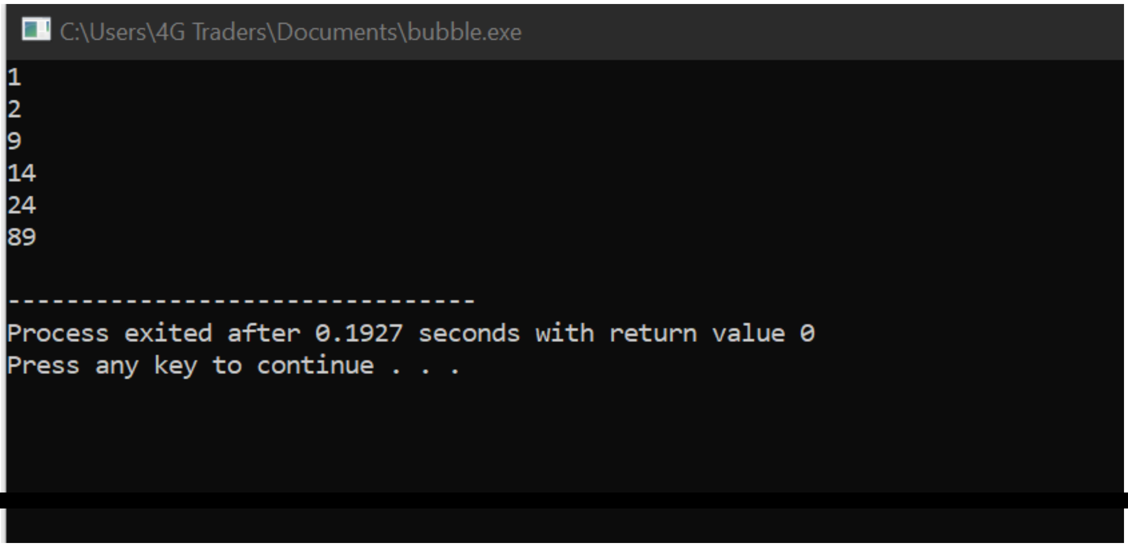

C:\Users\4G Traders\Documents\bubble.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit
(globals)
Project Classes Debug
Untitled1.cpp Untitled2.cpp [*] Untitled3.cpp [*] Untitled4.cpp Untitled1.cpp [*] Untitled2 Untitled3.cpp [*] bubble.

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

int main() {
    int temp, i, j;
    int x[6] = {14, 2, 9, 1, 89, 24};
    for (int i = 0; i <= 5; i++) {
        for (int j = i + 1; j <= 5; j++) {
            if (x[i] > x[j]) {
                temp = x[i];
                x[i] = x[j];
                x[j] = temp;
            }
        }
    }
    for (int k = 0; k <= 5; k++)
        cout << x[k] << endl;

    return 0;
}
```

Logic Explanation:

- The program initializes an array x with 6 integers.
- It uses the Bubble Sort algorithm to sort the elements of the array in ascending order.
- The outer loop iterates through each element of the array.
- The inner loop compares adjacent elements and swaps them if they are in the wrong order.
- After completing the Bubble Sort, the program displays the sorted array.

Output:



```
1
2
9
14
24
89

--------------------------------
Process exited after 0.1927 seconds with return value 0
Press any key to continue . . .
```

Output Explanation:

- The array {14, 2, 9, 1, 89, 24} is sorted in ascending order using the Bubble Sort algorithm.
- The sorted array is then displayed.
- The output represents the elements of the array after the sorting process.
- In this specific case, the sorted array is {1, 2, 9, 14, 24, 89}.

6. Solve any Aerospace/Real Life Problem using C++ Programming. 15

Let's consider a real-life problem of calculating and tracking expenses for a personal budget. The program will allow the user to input various expenses and incomes, and it will then calculate the remaining budget.

Introduction:

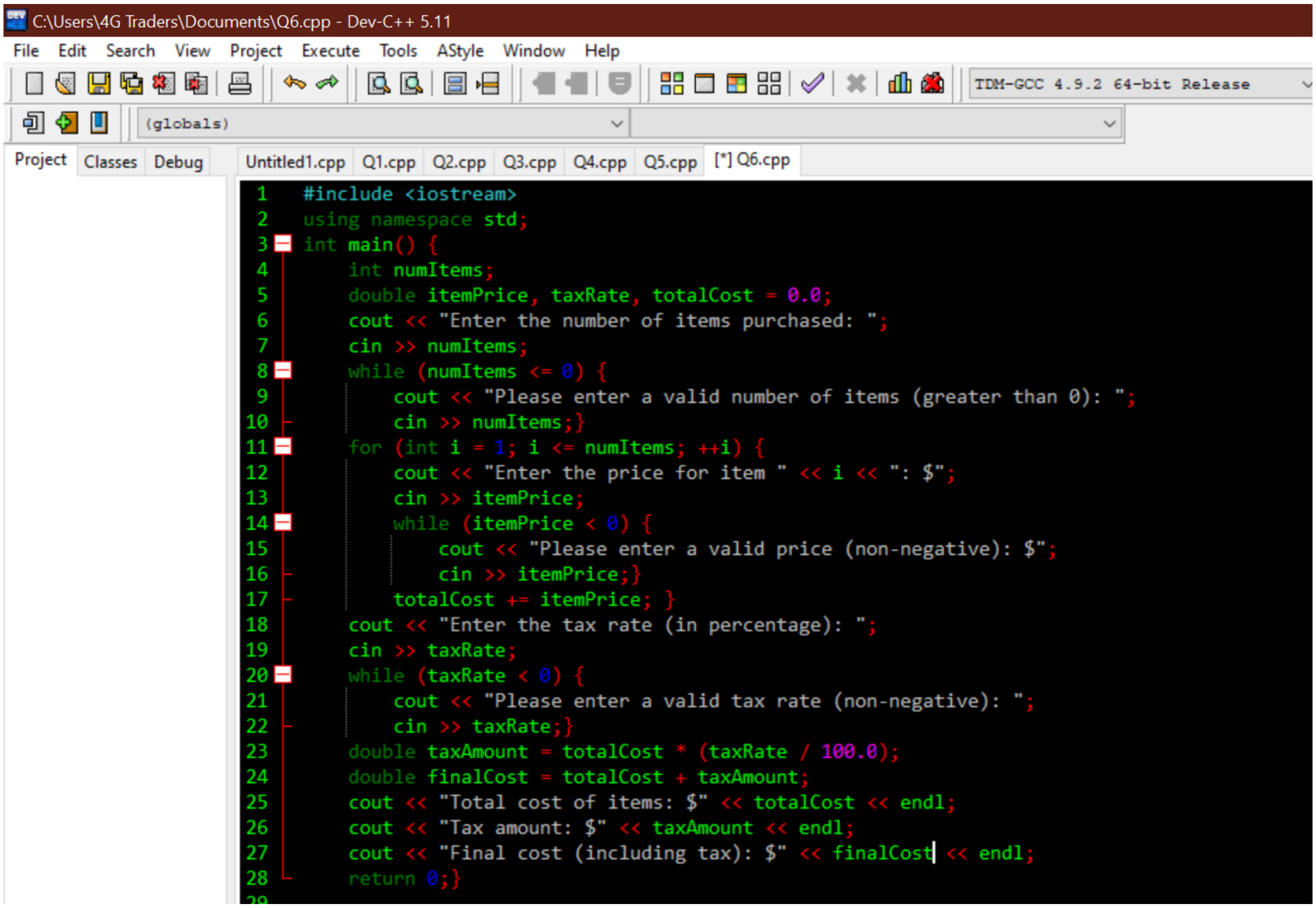
Embarking on a journey to solve real-life problems using C++ programming is an exciting and empowering endeavor. In our daily lives, we encounter numerous challenges that can be addressed and optimized through the application of computational solutions. For this project, I have chosen to tackle a common and practical issue that many individuals face: budget management.

Objective:

The objective of this program is to assist users in managing their expenses while considering tax implications. The program takes user input for the number of items purchased, their prices, and the tax rate. It then calculates and displays the total cost of the items, the tax amount, and the final cost, including tax.

Program:

```
#include <iostream>
using namespace std;
int main() {
    int numItems;
    double itemPrice, taxRate, totalCost = 0.0;
    cout << "Enter the number of items purchased: ";
    cin >> numItems;
    while (numItems <= 0) {
        cout << "Please enter a valid number of items (greater than 0): ";
        cin >> numItems;}
    for (int i = 1; i <= numItems; ++i) {
        cout << "Enter the price for item " << i << ": $";
        cin >> itemPrice;
        while (itemPrice < 0) {
            cout << "Please enter a valid price (non-negative): $";
            cin >> itemPrice;}
        totalCost += itemPrice; }
    cout << "Enter the tax rate (in percentage): ";
    cin >> taxRate;
    while (taxRate < 0) {
        cout << "Please enter a valid tax rate (non-negative): ";
        cin >> taxRate;}
    double taxAmount = totalCost * (taxRate / 100.0);
    double finalCost = totalCost + taxAmount;
    cout << "Total cost of items: $" << totalCost << endl;
    cout << "Tax amount: $" << taxAmount << endl;
    cout << "Final cost (including tax): $" << finalCost << endl;
    return 0;}
```


C:\Users\4G Traders\Documents\Q6.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Untitled1.cpp Q1.cpp Q2.cpp Q3.cpp Q4.cpp Q5.cpp [*] Q6.cpp

```
#include <iostream>
using namespace std;
int main() {
    int numItems;
    double itemPrice, taxRate, totalCost = 0.0;
    cout << "Enter the number of items purchased: ";
    cin >> numItems;
    while (numItems <= 0) {
        cout << "Please enter a valid number of items (greater than 0): ";
        cin >> numItems;}
    for (int i = 1; i <= numItems; ++i) {
        cout << "Enter the price for item " << i << ": $";
        cin >> itemPrice;
        while (itemPrice < 0) {
            cout << "Please enter a valid price (non-negative): $";
            cin >> itemPrice;}
        totalCost += itemPrice; }
    cout << "Enter the tax rate (in percentage): ";
    cin >> taxRate;
    while (taxRate < 0) {
        cout << "Please enter a valid tax rate (non-negative): ";
        cin >> taxRate;}
    double taxAmount = totalCost * (taxRate / 100.0);
    double finalCost = totalCost + taxAmount;
    cout << "Total cost of items: $" << totalCost << endl;
    cout << "Tax amount: $" << taxAmount << endl;
    cout << "Final cost (including tax): $" << finalCost << endl;
    return 0;}
```

Explanation:

User Input Section:

- The program starts by prompting the user to enter the number of items purchased (numItems).
- It uses a while loop to ensure that the user enters a valid number (greater than 0) by displaying a prompt until a valid input is provided

Item Details Input Section:

- The program then enters a for loop to gather details about each item.
- For each item, it prompts the user to enter the price (itemPrice) and validates that the entered price is non-negative.

Tax Rate Input Section:

After collecting item details, the program prompts the user to enter the tax rate (taxRate) and validates that the entered rate is non-negative.

Calculations Section:

- The program calculates the total cost of the items (totalCost) by summing up the individual item prices.
- It calculates the tax amount (taxAmount) based on the total cost and the entered tax rate.
- The final cost including tax (finalCost) is then computed.

Explanation of output:

User Input Section:

The program starts by prompting the user to enter the number of items purchased (numItems).
It uses a while loop to ensure that the user enters a valid number (greater than 0) by displaying a prompt until a valid input is provided.

Item Details Input Section:

The program then enters a for loop to gather details about each item.
For each item, it prompts the user to enter the price (itemPrice) and validates that the entered price is non-negative.

Tax Rate Input Section:

After collecting item details, the program prompts the user to enter the tax rate (taxRate) and validates that the entered rate is non-negative.

Calculations Section:

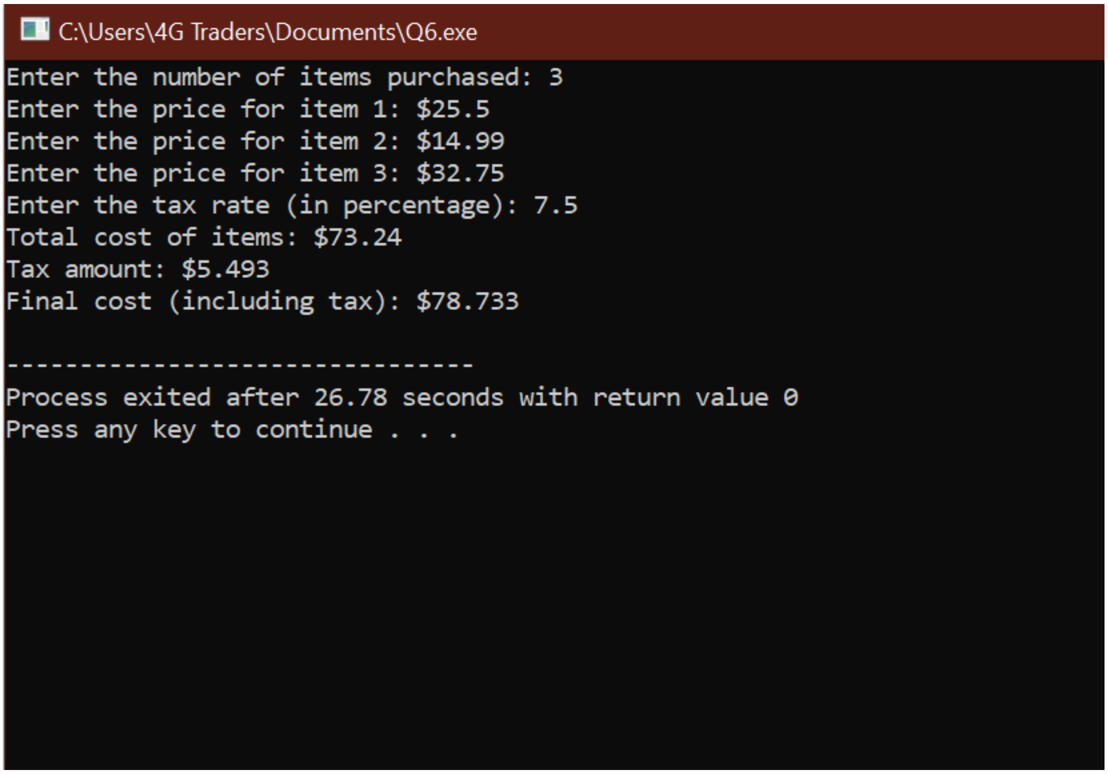
The program calculates the total cost of the items (totalCost) by summing up the individual item prices.
It calculates the tax amount (taxAmount) based on the total cost and the entered tax rate.
The final cost including tax (finalCost) is then computed.

Output Section:

Finally, the program displays the results.
It shows the total cost of items, the tax amount, and the final cost (including tax).

Example Output:

In this example, the user inputs three items with their respective prices, and a tax rate of 7.5%. The program then calculates and displays the total cost of items, tax amount, and final cost.



```
Enter the number of items purchased: 3
Enter the price for item 1: $25.5
Enter the price for item 2: $14.99
Enter the price for item 3: $32.75
Enter the tax rate (in percentage): 7.5
Total cost of items: $73.24
Tax amount: $5.493
Final cost (including tax): $78.733

--------------------------------
Process exited after 26.78 seconds with return value 0
Press any key to continue . . .
```