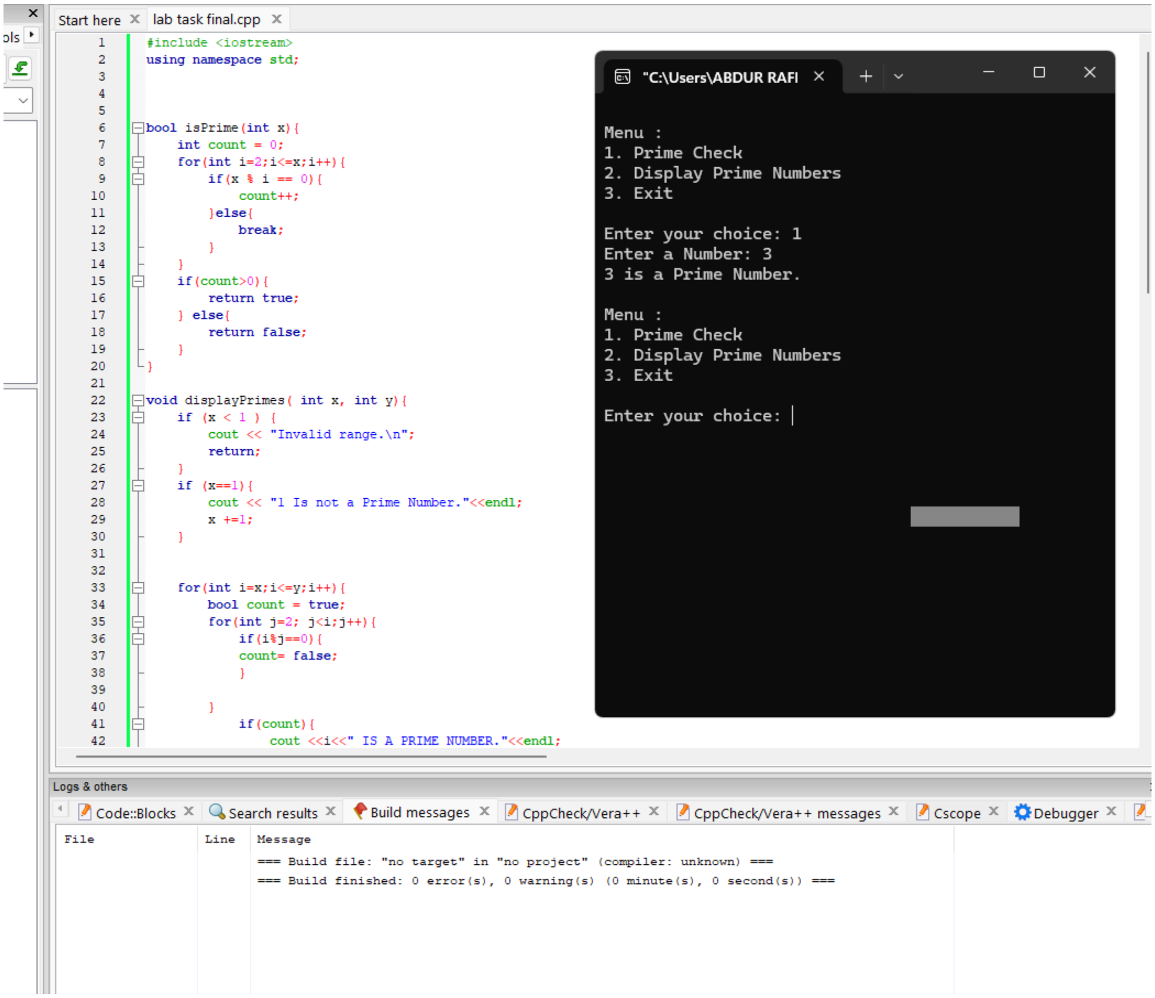
## Name:Md.Abdur Rafi

## Id:24-57874-2

## SL Number:15

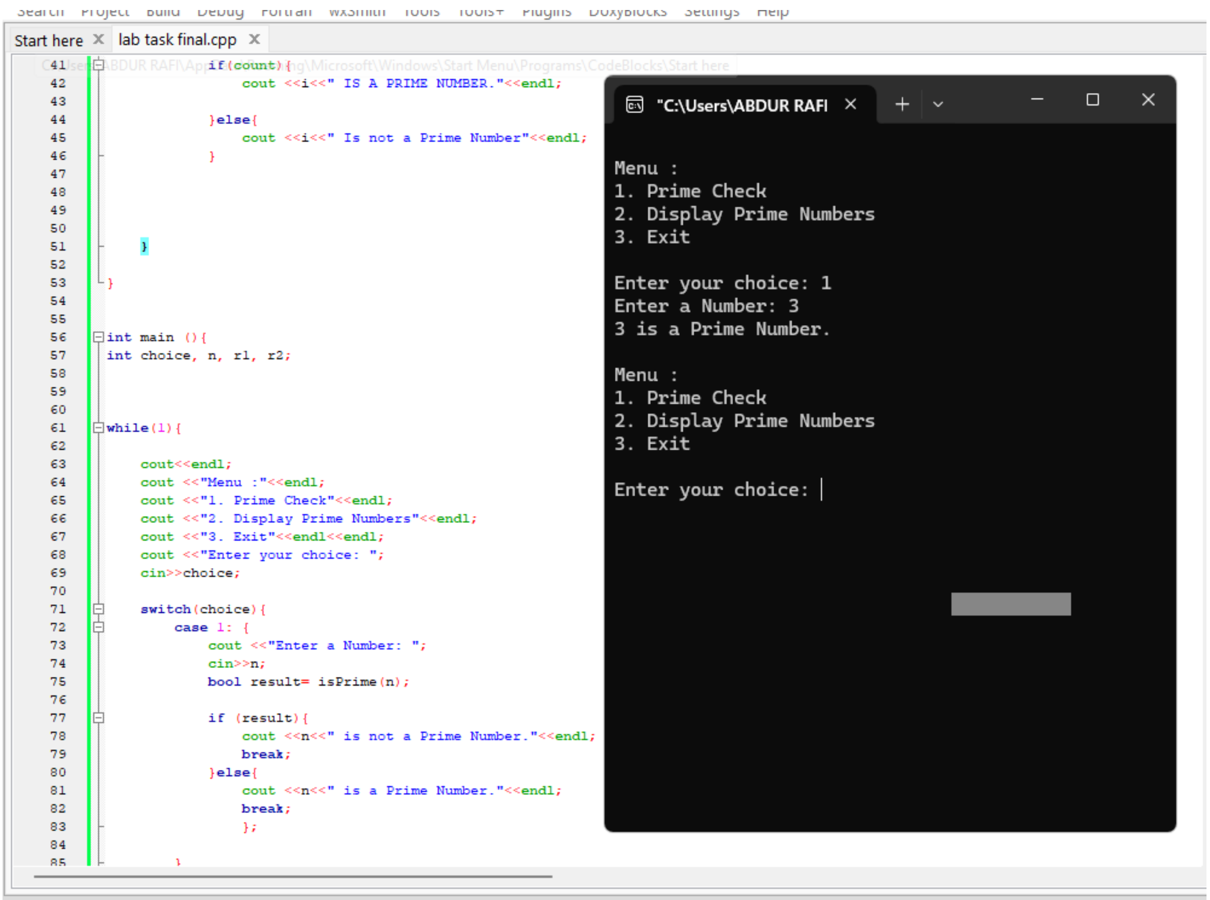
**Question: 1. Imagine you are tasked with developing an educational module for high school students exploring number theory as part of an educational software suite. This module is designed to efficiently teach the concepts of identifying a single prime number and listing all prime numbers within a specified range, enhancing students' understanding of prime numbers through interactive learning. Your assignment is to create a C++ program that includes a function named isPrime to check if a number is prime, and another function named displayPrimes to print all prime numbers between two given intervals. The main () function should allow users to choose between checking the primality of a single number or displaying all prime numbers within a specified range, prompting for a single number or two interval defining numbers accordingly, and executing the relevant function based on the user's choice. This hands-on tool aims to provide a comprehensive learning experience by allowing students to engage directly with the material through practical application and verification of mathematical theories.**



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



bool isPrime(int x){
    int count = 0;
    for(int i=2;i<=x;i++){
        if(x % i == 0){
            count++;
        }else{
            break;
        }
    }
    if(count>0){
        return true;
    } else{
        return false;
    }
}

void displayPrimes( int x, int y){
    if (x < 1 ) {
        cout << "Invalid range.\n";
        return;
    }
    if (x==1){
        cout << "1 Is not a Prime Number."<<endl;
        x +=1;
    }


    for(int i=x;i<=y;i++){
        bool count = true;
        for(int j=2; j<i;j++){
            if(i%j==0){
            count= false;
            }

        }
            if(count){
                cout <<i<<" IS A PRIME NUMBER."<<endl;
```

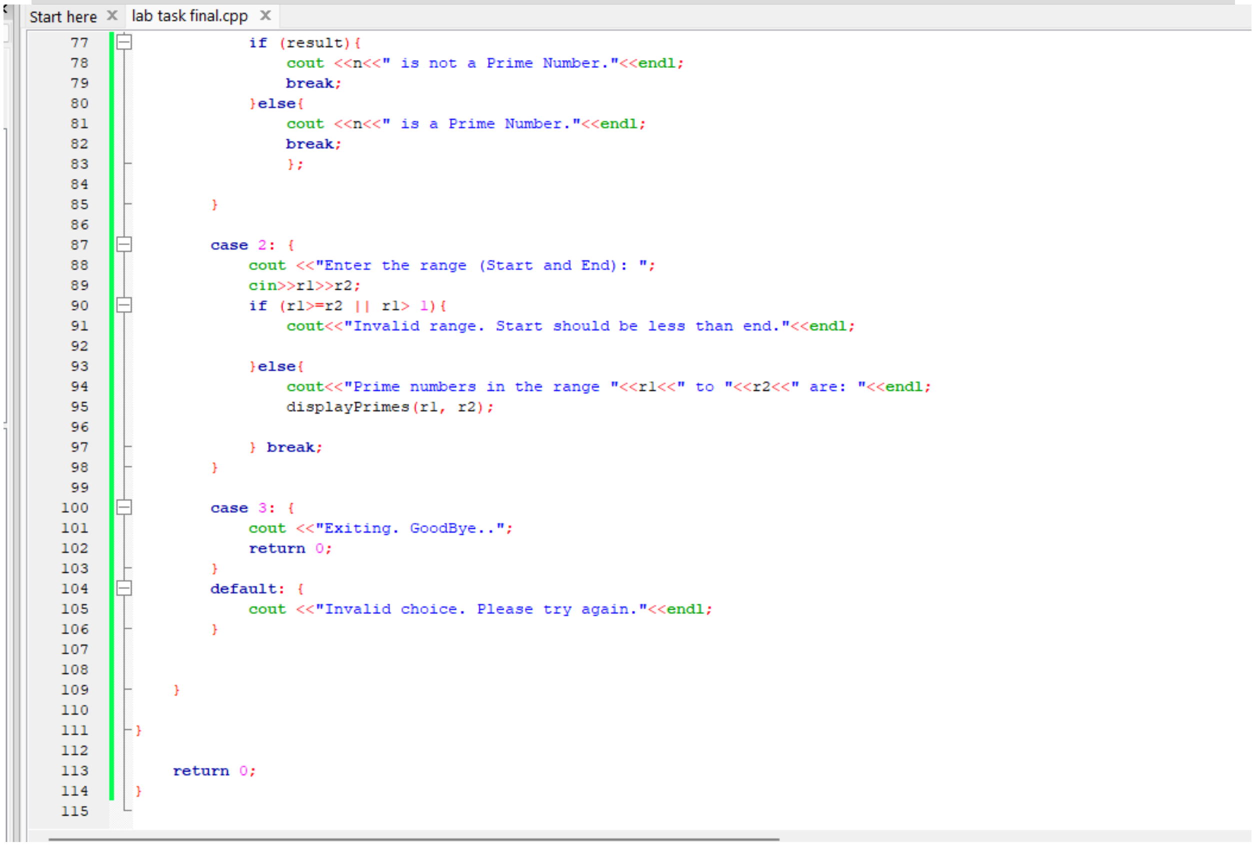
```
Menu :
1. Prime Check
2. Display Prime Numbers
3. Exit

Enter your choice: 1
Enter a Number: 3
3 is a Prime Number.

Menu :
1. Prime Check
2. Display Prime Numbers
3. Exit

Enter your choice:
```

```
File            Line    Message
                        === Build file: "no target" in "no project" (compiler: unknown) ===
                        === Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
```


Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
lab task final.cpp
"C:\Users\ABDUR RAFI

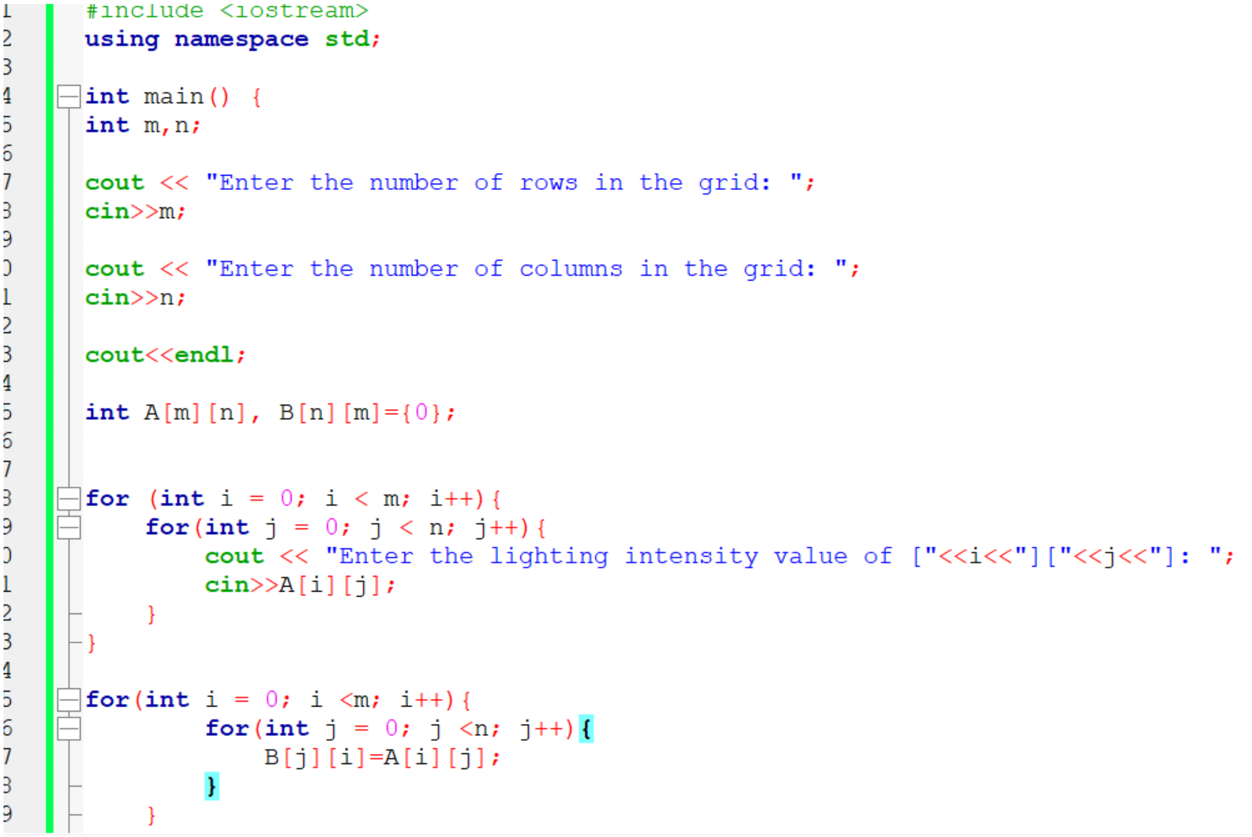
```
            if(count){
                cout <<i<<" IS A PRIME NUMBER."<<endl;

            }else{
                cout <<i<<" Is not a Prime Number"<<endl;
            }




    }

}


int main (){
int choice, n, r1, r2;



while(1){

    cout<<endl;
    cout <<"Menu :"<<endl;
    cout <<"1. Prime Check"<<endl;
    cout <<"2. Display Prime Numbers"<<endl;
    cout <<"3. Exit"<<endl<<endl;
    cout <<"Enter your choice: ";
    cin>>choice;

    switch(choice){
        case 1: {
            cout <<"Enter a Number: ";
            cin>>n;
            bool result= isPrime(n);

            if (result){
                cout <<n<<" is not a Prime Number."<<endl;
                break;
            }else{
                cout <<n<<" is a Prime Number."<<endl;
                break;
                };

        }
```

```
Menu :
1. Prime Check
2. Display Prime Numbers
3. Exit

Enter your choice: 1
Enter a Number: 3
3 is a Prime Number.

Menu :
1. Prime Check
2. Display Prime Numbers
3. Exit

Enter your choice: 
```

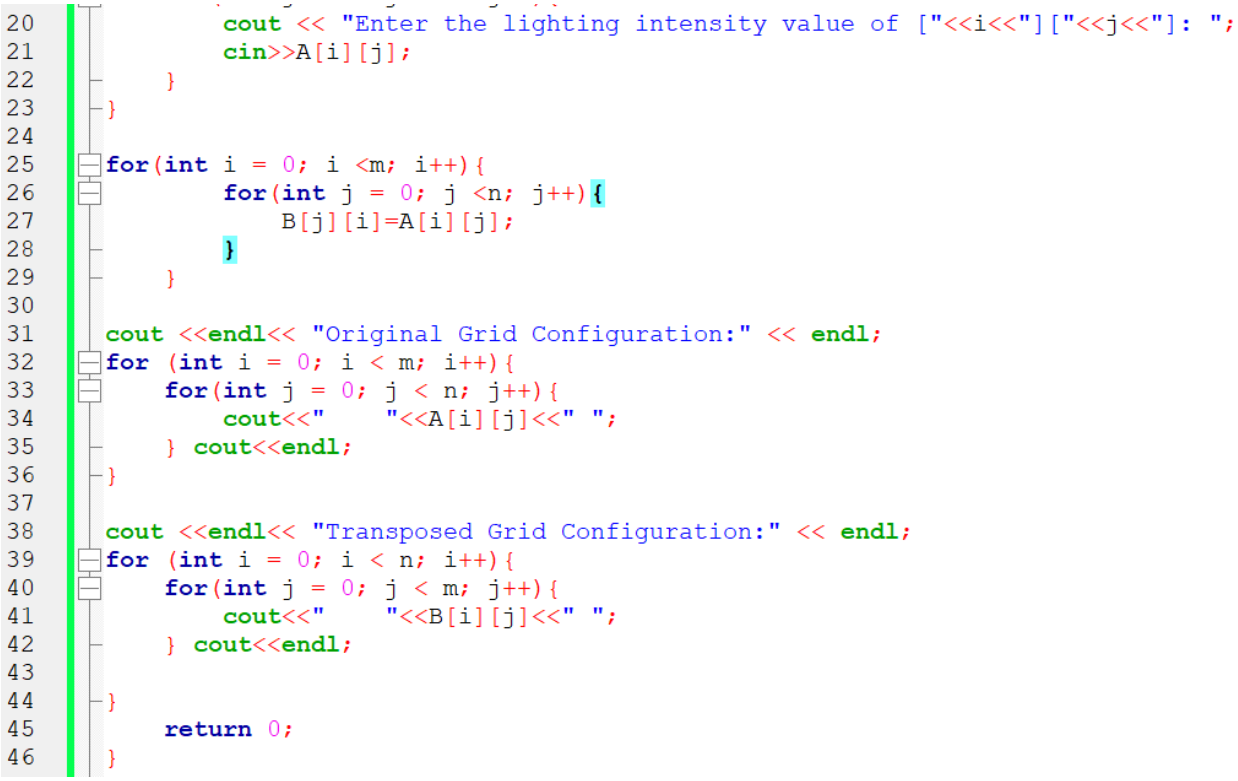


```
            if (result){
                cout <<n<<" is not a Prime Number."<<endl;
                break;
            }else{
                cout <<n<<" is a Prime Number."<<endl;
                break;
                };

        }

        case 2: {
            cout <<"Enter the range (Start and End): ";
            cin>>r1>>r2;
            if (r1>=r2 || r1> 1){
                cout<<"Invalid range. Start should be less than end."<<endl;

            }else{
                cout<<"Prime numbers in the range "<<r1<<" to "<<r2<<" are: "<<endl;
                displayPrimes(r1, r2);

            } break;
        }

        case 3: {
            cout <<"Exiting. GoodBye..";
            return 0;
        }
        default: {
            cout <<"Invalid choice. Please try again."<<endl;
        }


    }

}

    return 0;
}
```

**Question 2: 2. Imagine you are an IT consultant for a modern art gallery. The gallery is planning an innovative exhibition where artworks are illuminated by a complex grid of adjustable LED lights. This grid is controlled by a computer program that adjusts the lighting based on the time of day and the crowd in the gallery. The director wants a special feature: to be able to flip the lighting pattern from rows to columns and vice versa to create different moods, essentially requiring a matrix transpose operation on the grid's configuration. Objective: Your task is to write a C++ program that simulates the transpose operation on the LED lighting grid's configuration matrix.Requirements:➢User Input for Grid Dimensions: Your program should begin by asking the user (gallery technician) forthe dimensions of the LED grid, i.e., the number of rows and columns.➢Grid Configuration Input: Next, prompt the user to enter the lighting intensity values (you can assumethese are integers for simplicity) for each cell of the grid, row by row. This represents the current configuration of the lighting grid. ➢Computing Transpose: Once the grid configuration is entered, your program should compute the transpose of this matrix, effectively simulating the switch in lighting patterns from rows to columns.➢Display Output: Finally, your program should display the original grid configuration and the transposed grid configuration. This will help the gallery technician visualize the change in lighting patterns.**

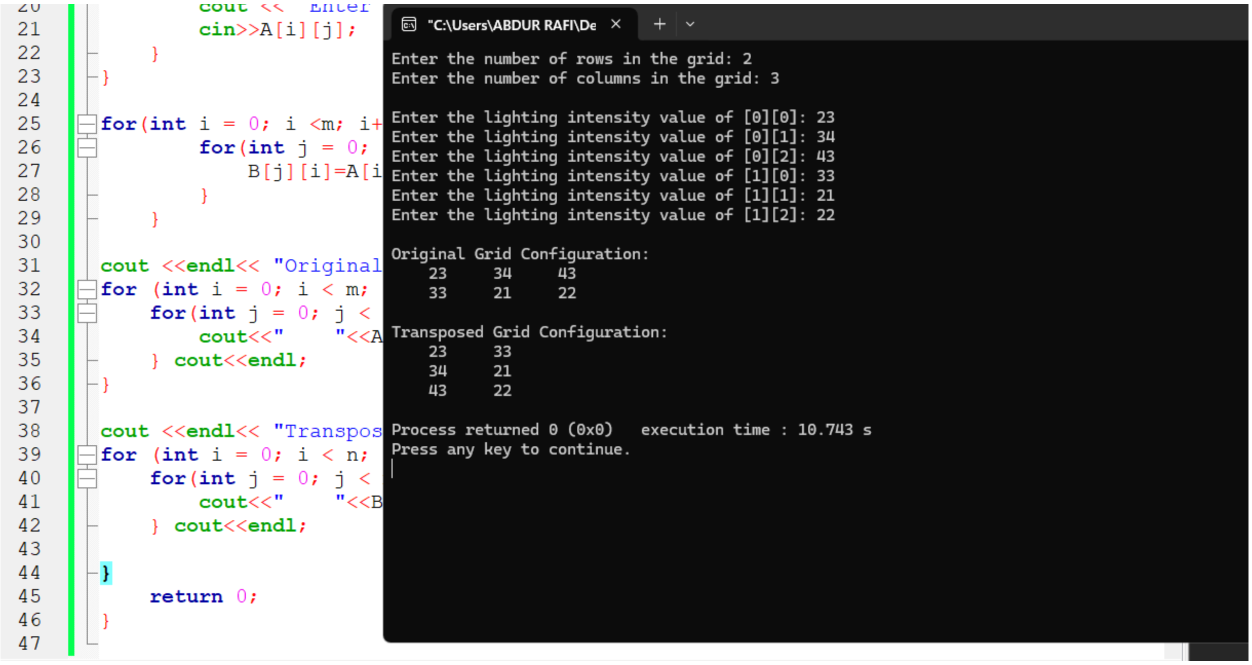


```cpp
#include <iostream>
using namespace std;

int main() {
int m,n;

cout << "Enter the number of rows in the grid: ";
cin>>m;

cout << "Enter the number of columns in the grid: ";
cin>>n;

cout<<endl;

int A[m][n], B[n][m]={0};


for  (int i = 0; i < m; i++){
    for(int j = 0; j < n; j++){
        cout << "Enter the lighting intensity value of ["<<i<<"]["<<j<<"]: ";
        cin>>A[i][j];
    }
}

for(int i = 0; i <m; i++){
        for(int j = 0; j <n; j++){
            B[j][i]=A[i][j];
        }
    }
```



```
        cout << "Enter the lighting intensity value of ["<<i<<"]["<<j<<"]: ";
        cin>>A[i][j];
    }
}

for(int i = 0; i <m; i++){
        for(int j = 0; j <n; j++){
            B[j][i]=A[i][j];
        }
    }

cout <<endl<< "Original Grid Configuration:" << endl;
for (int i = 0; i < m; i++){
    for(int j = 0; j < n; j++){
        cout<<"     "<<A[i][j]<<" ";
    } cout<<endl;
}

cout <<endl<< "Transposed Grid Configuration:" << endl;
for (int i = 0; i < n; i++){
    for(int j = 0; j < m; j++){
        cout<<"     "<<B[i][j]<<" ";
    } cout<<endl;

}

    return 0;
}
```

**Output:**

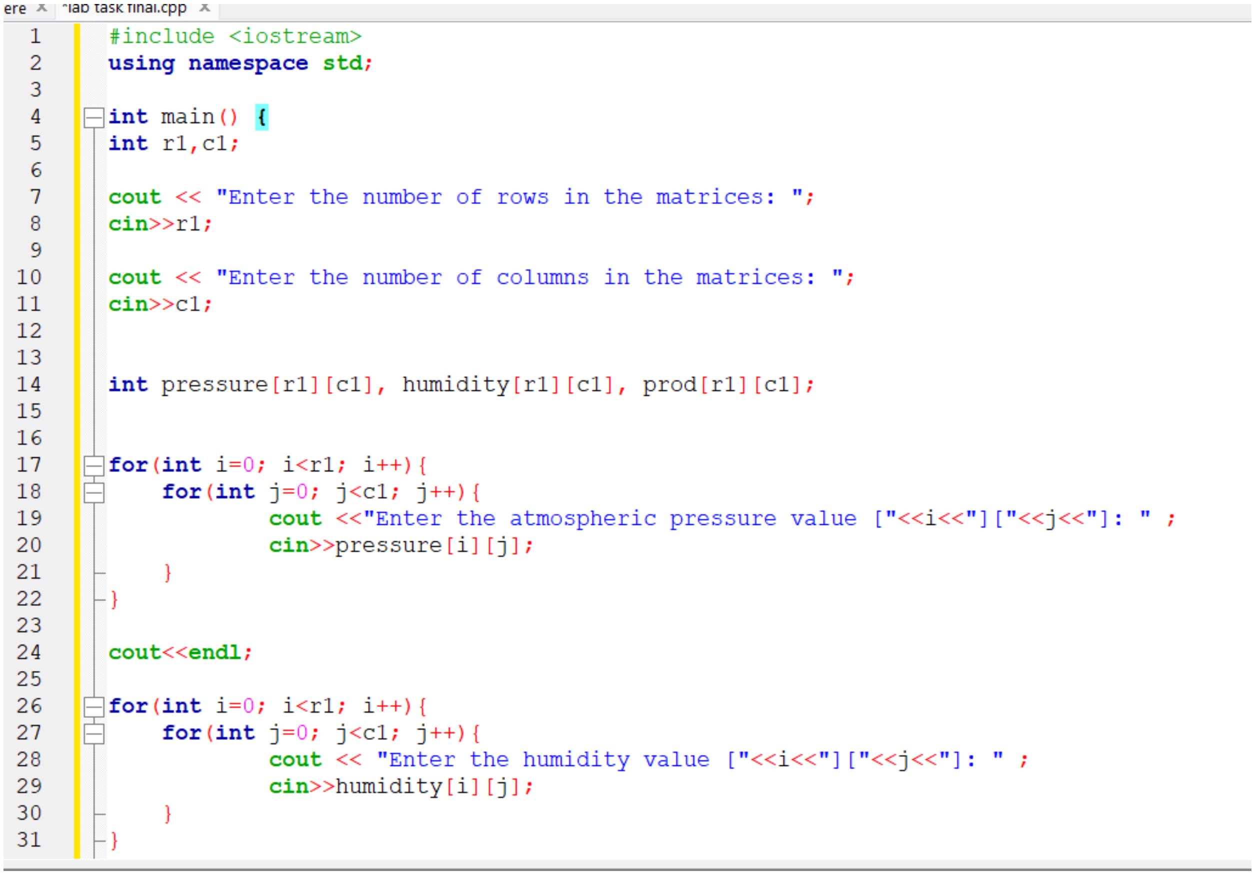


```
        cout << "Enter
        cin>>A[i][j];
    }
}

for(int i = 0; i <m; i+
        for(int j = 0;
            B[j][i]=A[i
        }
    }

cout <<endl<< "Original
for (int i = 0; i < m;
    for(int j = 0; j <
        cout<<"     "<<A
    } cout<<endl;
}

cout <<endl<< "Transpos
for (int i = 0; i < n;
    for(int j = 0; j <
        cout<<"     "<<B
    } cout<<endl;

}
    return 0;
}
```

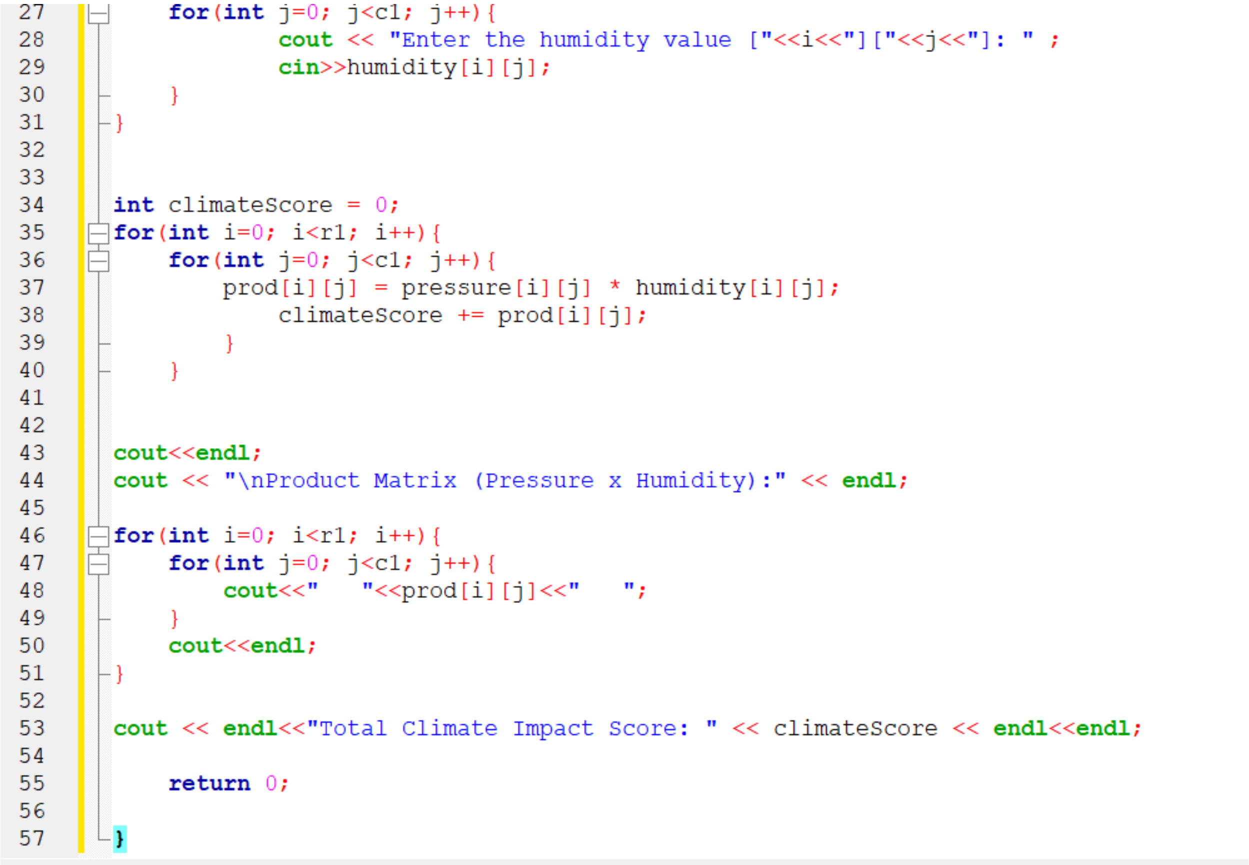
```
Enter the number of rows in the grid: 2
Enter the number of columns in the grid: 3

Enter the lighting intensity value of [0][0]: 23
Enter the lighting intensity value of [0][1]: 34
Enter the lighting intensity value of [0][2]: 43
Enter the lighting intensity value of [1][0]: 33
Enter the lighting intensity value of [1][1]: 21
Enter the lighting intensity value of [1][2]: 22

Original Grid Configuration:
     23      34      43
     33      21      22

Transposed Grid Configuration:
     23      33
     34      21
     43      22

Process returned 0 (0x0)   execution time : 10.743 s
Press any key to continue.
```

**Quation: 3. You are working on a project that simulates a small section of a weather prediction model. The project requires you to work with two specific types of data matrices: one representing atmospheric pressure values across different geographical zones, and the other representing humidity levels in the same zones. Your task is to develop a C++ program that uses multi-dimensional arrays to accomplish two main objectives. First, you must calculate the product of these two matrices, if each cell in the product matrix represents the combined effect of pressure and humidity on the local climate. Secondly, you must find the sum of all the elements in the resulting matrix to estimate the overall climate impact score for the region under study. The program should be interactive, allowing users to input the values for both matrices, and then display both the product matrix and the overall climate impact score. This will help your team to dynamically analyze the influence of different atmospheric conditions on the climate model**

**Input:**

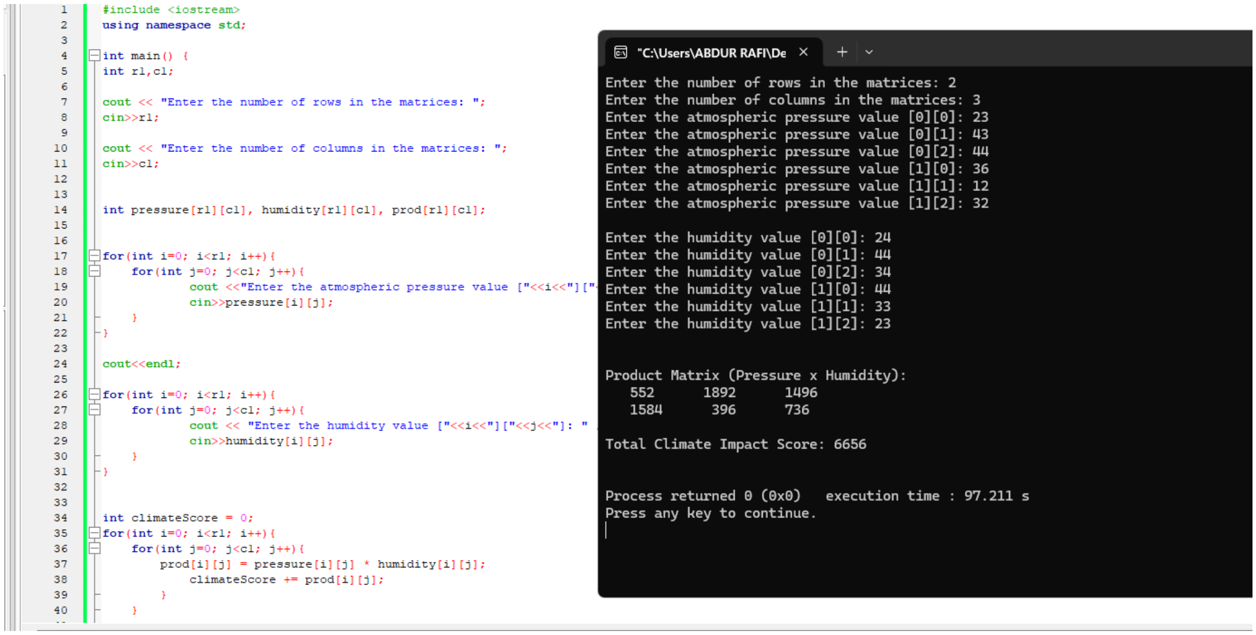


```cpp
#include <iostream>
using namespace std;

int main() {
int r1,c1;

cout << "Enter the number of rows in the matrices: ";
cin>>r1;

cout << "Enter the number of columns in the matrices: ";
cin>>c1;


int pressure[r1][c1], humidity[r1][c1], prod[r1][c1];


for(int i=0; i<r1; i++){
    for(int j=0; j<c1; j++){
            cout <<"Enter the atmospheric pressure value ["<<i<<"]["<<j<<"]: " ;
            cin>>pressure[i][j];
    }
}

cout<<endl;

for(int i=0; i<r1; i++){
    for(int j=0; j<c1; j++){
            cout << "Enter the humidity value ["<<i<<"]["<<j<<"]: " ;
            cin>>humidity[i][j];
    }
}
```



```
    for(int j=0; j<cl; j++){
            cout << "Enter the humidity value ["<<i<<"]["<<j<<"]: " ;
            cin>>humidity[i][j];
    }
}


int climateScore = 0;
for(int i=0; i<rl; i++){
    for(int j=0; j<cl; j++){
        prod[i][j] = pressure[i][j] * humidity[i][j];
            climateScore += prod[i][j];
        }
    }


cout<<endl;
cout << "\nProduct Matrix (Pressure x Humidity):" << endl;

for(int i=0; i<rl; i++){
    for(int j=0; j<cl; j++){
        cout<<"   "<<prod[i][j]<<"    ";
    }
    cout<<endl;
}

cout << endl<<"Total Climate Impact Score: " << climateScore << endl<<endl;

    return 0;

}
```

## Output:

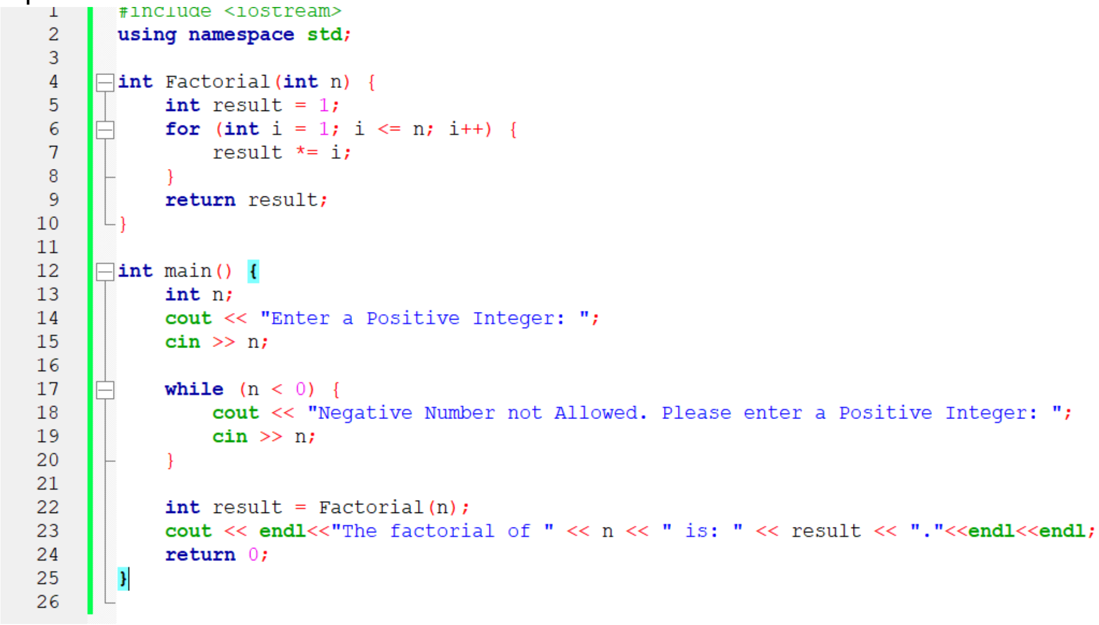


```
#include <iostream>
using namespace std;

int main() {
int rl,cl;

cout << "Enter the number of rows in the matrices: ";
cin>>rl;

cout << "Enter the number of columns in the matrices: ";
cin>>cl;


int pressure[rl][cl], humidity[rl][cl], prod[rl][cl];


for(int i=0; i<rl; i++){
    for(int j=0; j<cl; j++){
            cout <<"Enter the atmospheric pressure value ["<<i<<"]["
            cin>>pressure[i][j];
    }
}

cout<<endl;

for(int i=0; i<rl; i++){
    for(int j=0; j<cl; j++){
            cout << "Enter the humidity value ["<<i<<"]["<<j<<"]: "
            cin>>humidity[i][j];
    }
}


int climateScore = 0;
for(int i=0; i<rl; i++){
    for(int j=0; j<cl; j++){
        prod[i][j] = pressure[i][j] * humidity[i][j];
            climateScore += prod[i][j];
        }
    }
```

```
Enter the number of rows in the matrices: 2
Enter the number of columns in the matrices: 3
Enter the atmospheric pressure value [0][0]: 23
Enter the atmospheric pressure value [0][1]: 43
Enter the atmospheric pressure value [0][2]: 44
Enter the atmospheric pressure value [1][0]: 36
Enter the atmospheric pressure value [1][1]: 12
Enter the atmospheric pressure value [1][2]: 32

Enter the humidity value [0][0]: 24
Enter the humidity value [0][1]: 44
Enter the humidity value [0][2]: 34
Enter the humidity value [1][0]: 44
Enter the humidity value [1][1]: 33
Enter the humidity value [1][2]: 23


Product Matrix (Pressure x Humidity):
   552      1892      1496
   1584      396      736

Total Climate Impact Score: 6656


Process returned 0 (0x0)   execution time : 97.211 s
Press any key to continue.
```

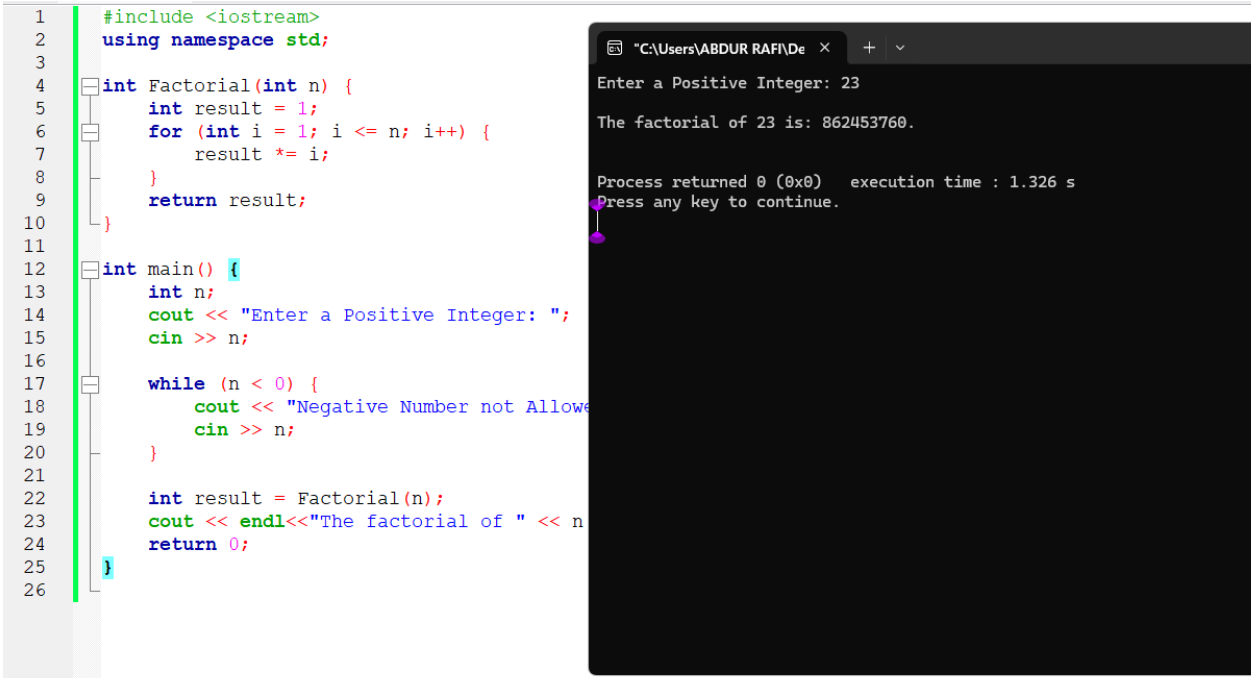
Question 5: You are tasked with developing an interactive feature for a mathematics tutoring website that demonstrates the concept of factorial, crucial for understanding permutations and combinations. Your assignment is to write a C++ program that includes a function named Factorial to compute the factorial of a number n (the product of all positive integers up to n). The program should prompt the user to input a positive integer, handle input errors like negative numbers or non-numeric inputs gracefully, and upon calculating the factorial, display the result along with an explanation of the factorial concept and its mathematical uses.

Input:



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

int Factorial(int n) {
    int result = 1;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

int main() {
    int n;
    cout << "Enter a Positive Integer: ";
    cin >> n;

    while (n < 0) {
        cout << "Negative Number not Allowed. Please enter a Positive Integer: ";
        cin >> n;
    }

    int result = Factorial(n);
    cout << endl<<"The factorial of " << n << " is: " << result << "."<<endl<<endl;
    return 0;
}
```

Output:



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

int Factorial(int n) {
    int result = 1;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

int main() {
    int n;
    cout << "Enter a Positive Integer: ";
    cin >> n;

    while (n < 0) {
        cout << "Negative Number not Allowe
        cin >> n;
    }

    int result = Factorial(n);
    cout << endl<<"The factorial of " << n
    return 0;
}
```

```
Enter a Positive Integer: 23

The factorial of 23 is: 862453760.


Process returned 0 (0x0)   execution time : 1.326 s
Press any key to continue.
```

Question: 6. In C++, design and implement a software program for a Bookstore Inventory Management System,

aimed at assisting a mid-sized bookstore in efficiently managing its book stocks and sales transactions. Your

task is to create a user-friendly system that allows bookstore staff to effectively oversee inventory and

customer purchases. Here are the specific functionalities you need to implement:

➢ Add New Book: This function should allow the user to input details for a new book,including ISBN,

title, author, genre, and price. The program should be able to handle up to 500 books. Implement error

handling to validate input formats and ensure data integrity.

➢ Display All Books: Provide a detailed list of all books in inventory, showing ISBN, title, author, genre,

and price. Use appropriate formatting to ensure readability and implement pagination to manage displays

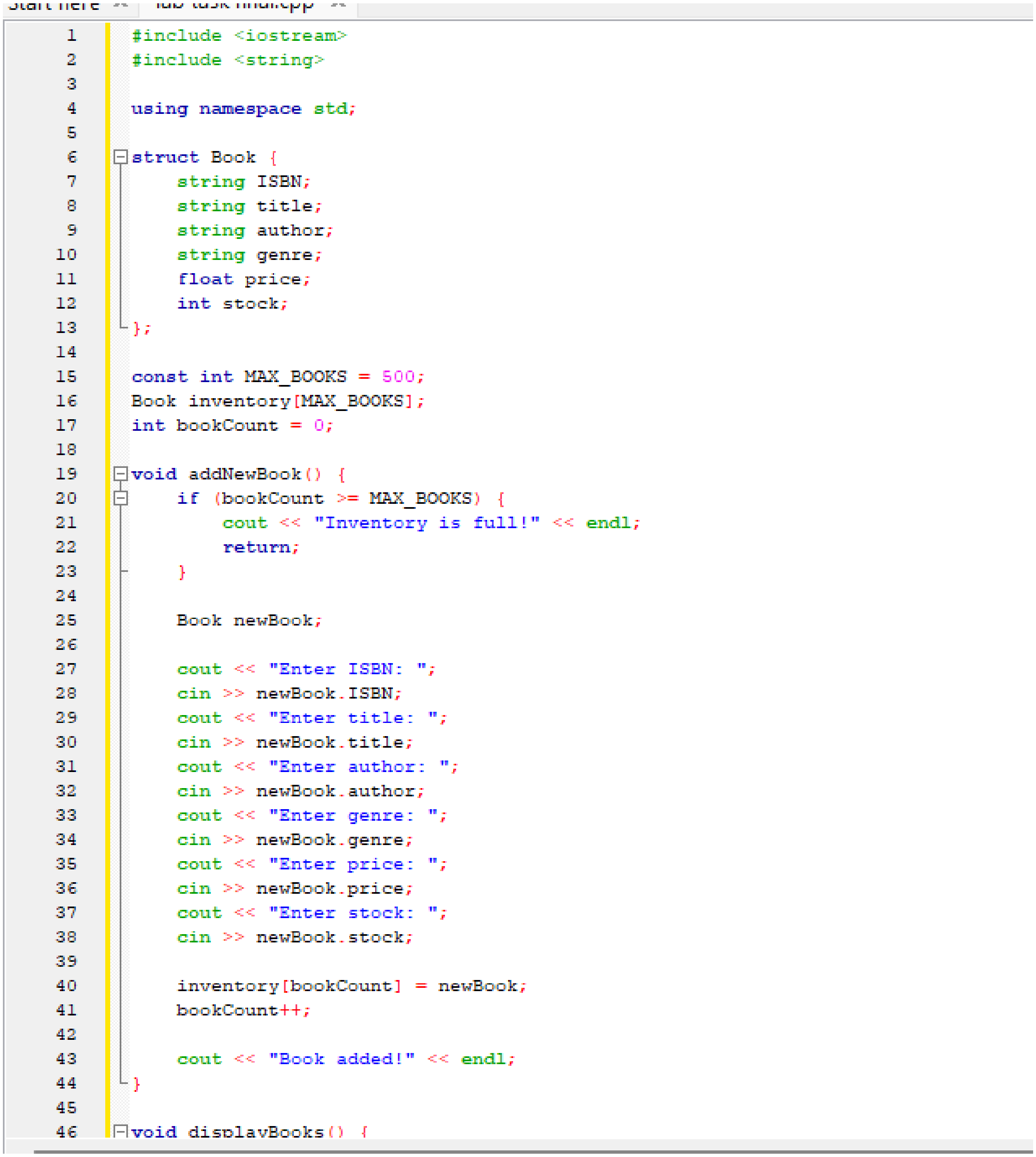
when the inventory list is extensive.

➢ Update Stock: Include a function to update the stock level for a given book. The user should enter the

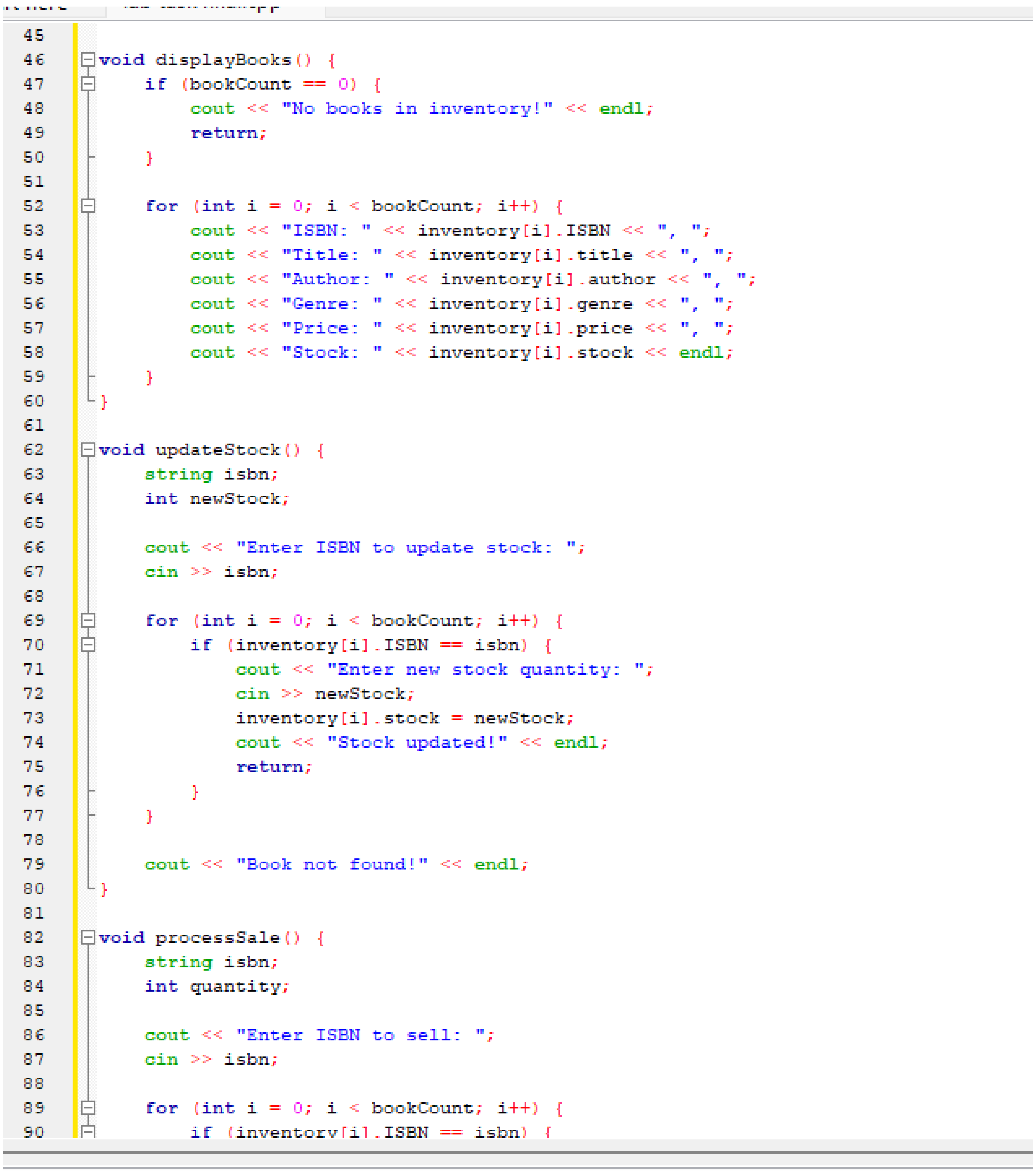
ISBN and the new stock level, with validations in place to ensure the book exists and the stock level is

Appropriate

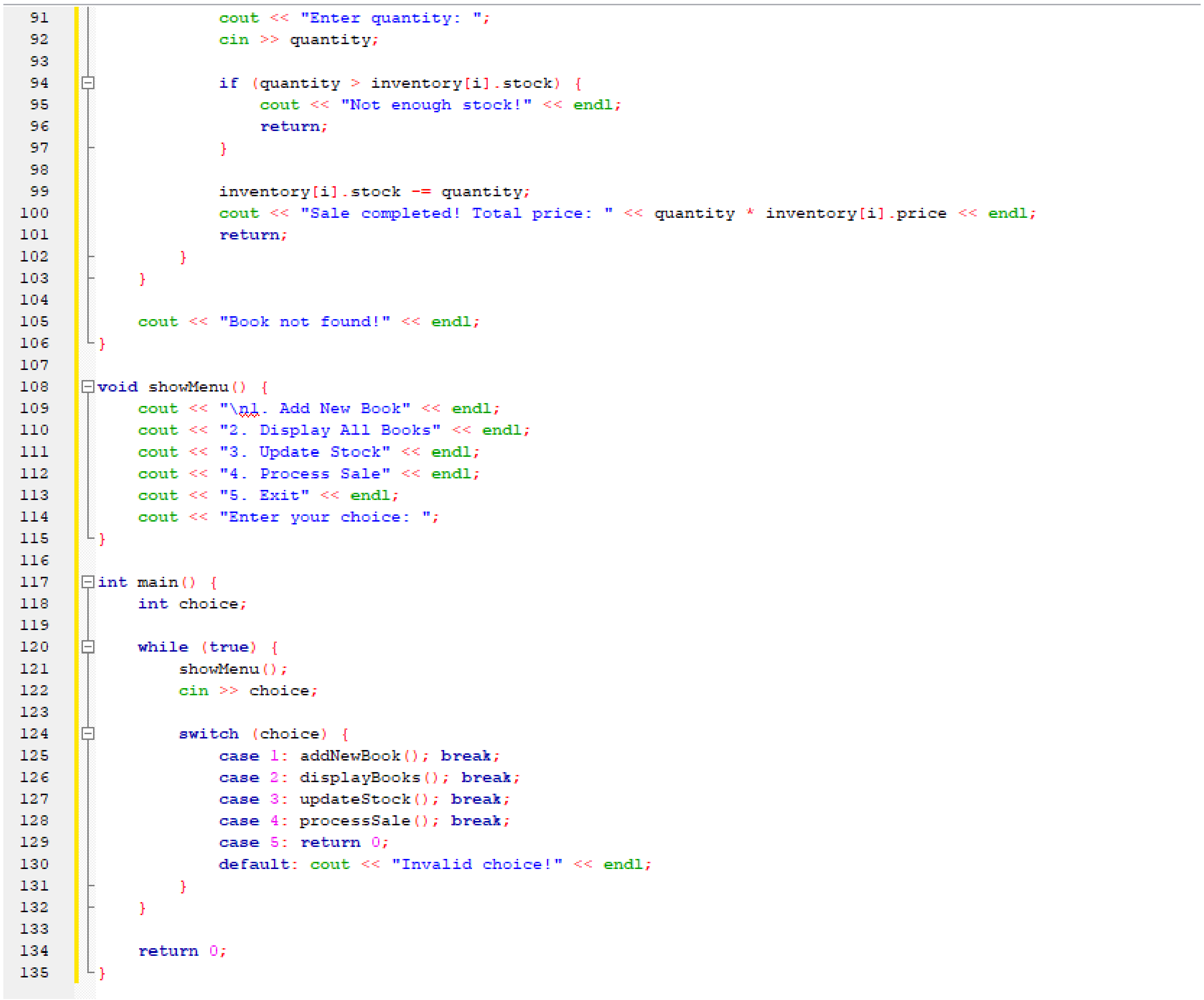
Input:



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

using namespace std;

struct Book {
    string ISBN;
    string title;
    string author;
    string genre;
    float price;
    int stock;
};

const int MAX_BOOKS = 500;
Book inventory[MAX_BOOKS];
int bookCount = 0;

void addNewBook() {
    if (bookCount >= MAX_BOOKS) {
        cout << "Inventory is full!" << endl;
        return;
    }

    Book newBook;

    cout << "Enter ISBN: ";
    cin >> newBook.ISBN;
    cout << "Enter title: ";
    cin >> newBook.title;
    cout << "Enter author: ";
    cin >> newBook.author;
    cout << "Enter genre: ";
    cin >> newBook.genre;
    cout << "Enter price: ";
    cin >> newBook.price;
    cout << "Enter stock: ";
    cin >> newBook.stock;

    inventory[bookCount] = newBook;
    bookCount++;

    cout << "Book added!" << endl;
}

void displayBooks() {
```



```cpp

void displayBooks() {
    if (bookCount == 0) {
        cout << "No books in inventory!" << endl;
        return;
    }

    for (int i = 0; i < bookCount; i++) {
        cout << "ISBN: " << inventory[i].ISBN << ", ";
        cout << "Title: " << inventory[i].title << ", ";
        cout << "Author: " << inventory[i].author << ", ";
        cout << "Genre: " << inventory[i].genre << ", ";
        cout << "Price: " << inventory[i].price << ", ";
        cout << "Stock: " << inventory[i].stock << endl;
    }
}

void updateStock() {
    string isbn;
    int newStock;

    cout << "Enter ISBN to update stock: ";
    cin >> isbn;

    for (int i = 0; i < bookCount; i++) {
        if (inventory[i].ISBN == isbn) {
            cout << "Enter new stock quantity: ";
            cin >> newStock;
            inventory[i].stock = newStock;
            cout << "Stock updated!" << endl;
            return;
        }
    }

    cout << "Book not found!" << endl;
}

void processSale() {
    string isbn;
    int quantity;

    cout << "Enter ISBN to sell: ";
    cin >> isbn;

    for (int i = 0; i < bookCount; i++) {
        if (inventory[i].ISBN == isbn) {
```



```cpp
            cout << "Enter quantity: ";
            cin >> quantity;

            if (quantity > inventory[i].stock) {
                cout << "Not enough stock!" << endl;
                return;
            }

            inventory[i].stock -= quantity;
            cout << "Sale completed! Total price: " << quantity * inventory[i].price << endl;
            return;
        }
    }

    cout << "Book not found!" << endl;
}

void showMenu() {
    cout << "\n1. Add New Book" << endl;
    cout << "2. Display All Books" << endl;
    cout << "3. Update Stock" << endl;
    cout << "4. Process Sale" << endl;
    cout << "5. Exit" << endl;
    cout << "Enter your choice: ";
}

int main() {
    int choice;

    while (true) {
        showMenu();
        cin >> choice;

        switch (choice) {
            case 1: addNewBook(); break;
            case 2: displayBooks(); break;
            case 3: updateStock(); break;
            case 4: processSale(); break;
            case 5: return 0;
            default: cout << "Invalid choice!" << endl;
        }
    }

    return 0;
}
```