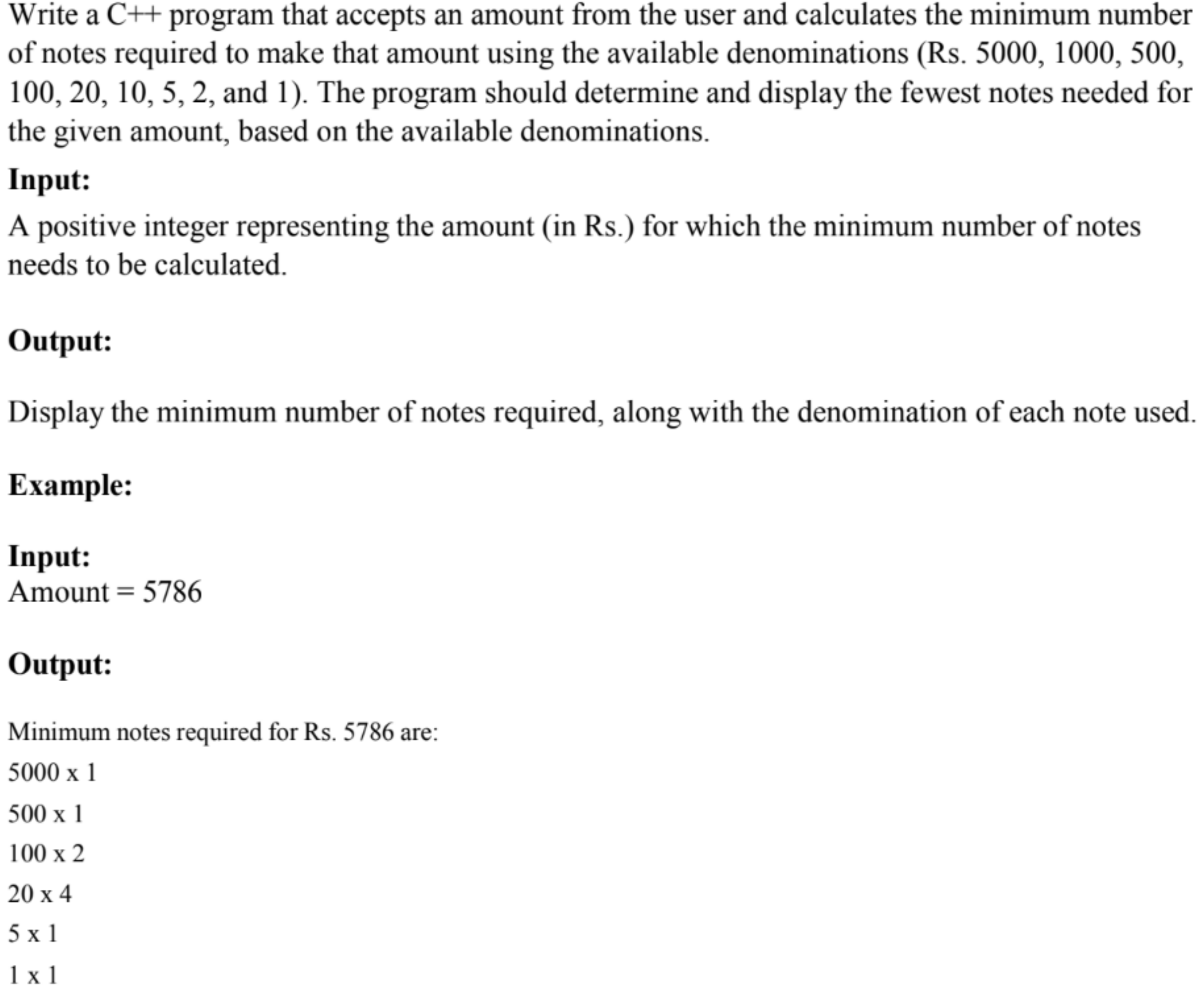
LAB 1

NAME PARI BATRA

ROLL NO 24K3115

SECTION bcs3a

TASK 1



Write a C++ program that accepts an amount from the user and calculates the minimum number of notes required to make that amount using the available denominations (Rs. 5000, 1000, 500, 100, 20, 10, 5, 2, and 1). The program should determine and display the fewest notes needed for the given amount, based on the available denominations.

Input:

A positive integer representing the amount (in Rs.) for which the minimum number of notes needs to be calculated.

Output:

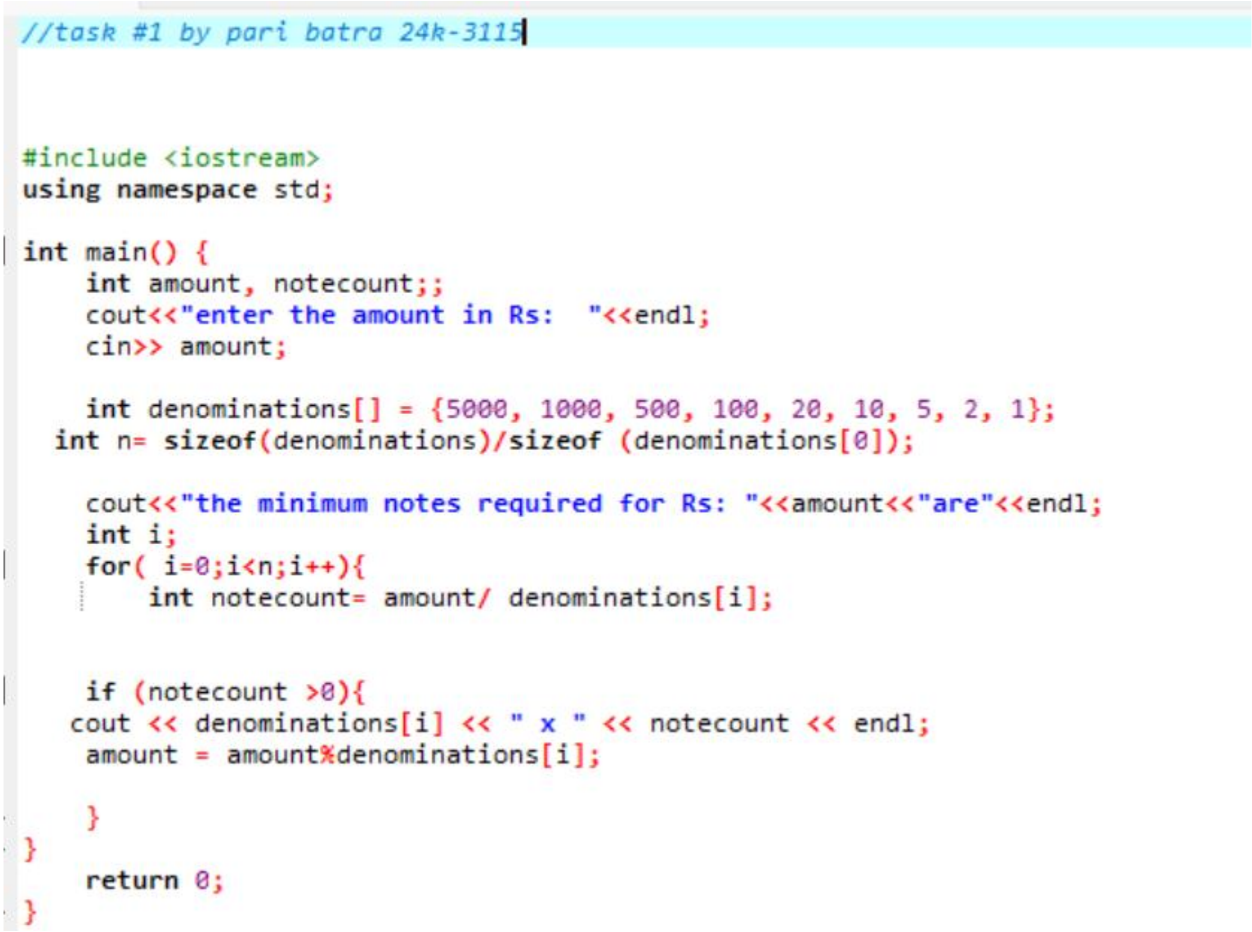
Display the minimum number of notes required, along with the denomination of each note used.

Example:

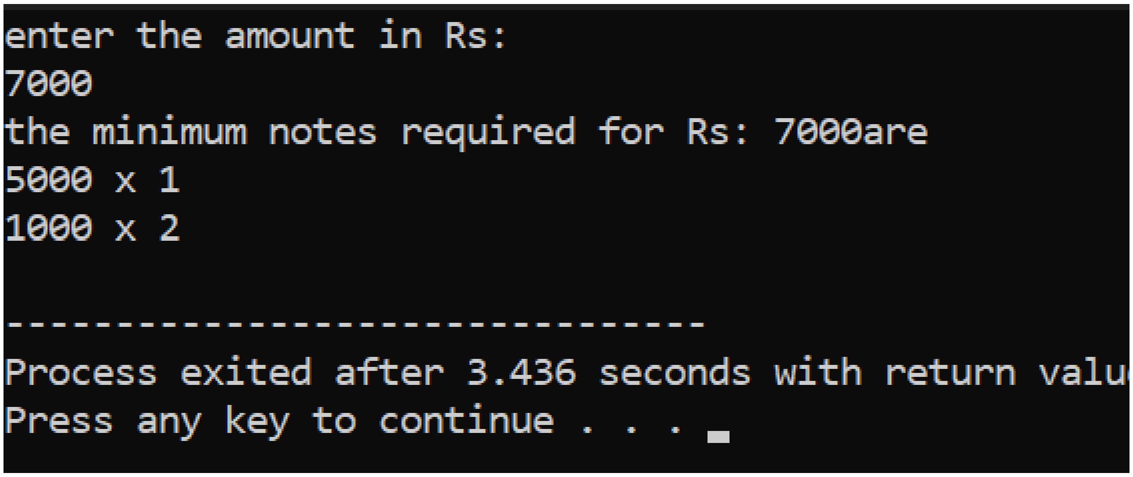
Input:
Amount = 5786

Output:

```
Minimum notes required for Rs. 5786 are:
5000 x 1
500 x 1
100 x 2
20 x 4
5 x 1
1 x 1
```



```
//task #1 by pari batra 24k-3115


#include <iostream>
using namespace std;

int main() {
    int amount, notecount;;
    cout<<"enter the amount in Rs:  "<<endl;
    cin>> amount;

    int denominations[] = {5000, 1000, 500, 100, 20, 10, 5, 2, 1};
  int n= sizeof(denominations)/sizeof (denominations[0]);

    cout<<"the minimum notes required for Rs: "<<amount<<"are"<<endl;
    int i;
    for( i=0;i<n;i++){
        int notecount= amount/ denominations[i];


    if (notecount >0){
   cout << denominations[i] << " x " << notecount << endl;
    amount = amount%denominations[i];

    }
}
    return 0;
}
```



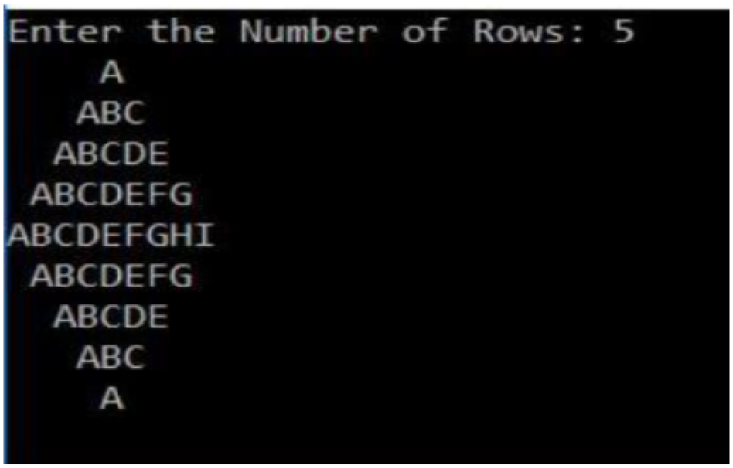
```
enter the amount in Rs:
7000
the minimum notes required for Rs: 7000are
5000 x 1
1000 x 2

--------------------------------
Process exited after 3.436 seconds with return valu
Press any key to continue . . .
```

TASK 2

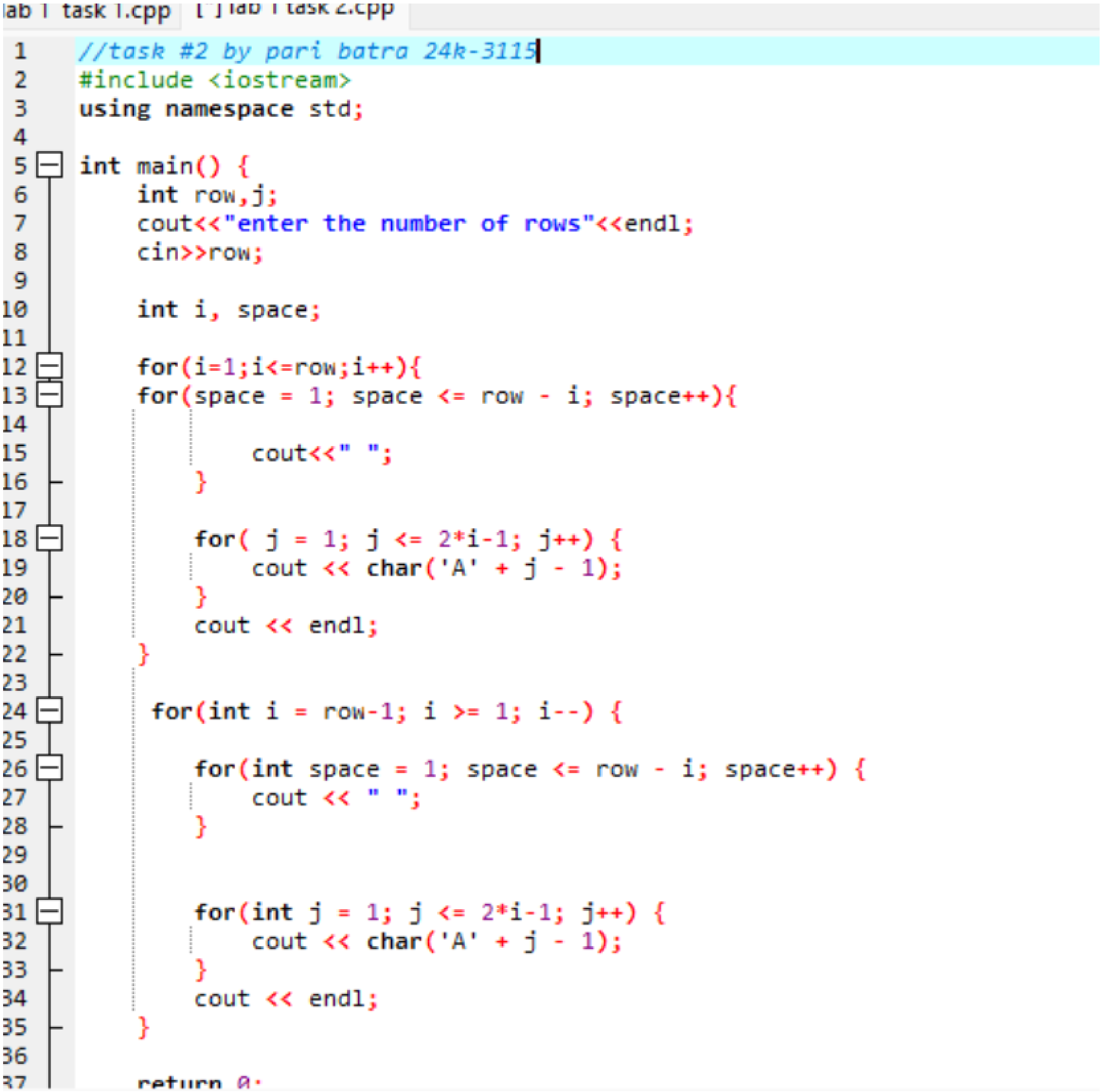
TASK # 02

Write a C++ program to input a number from user and print a diamond pattern of alphabet characters according to input.

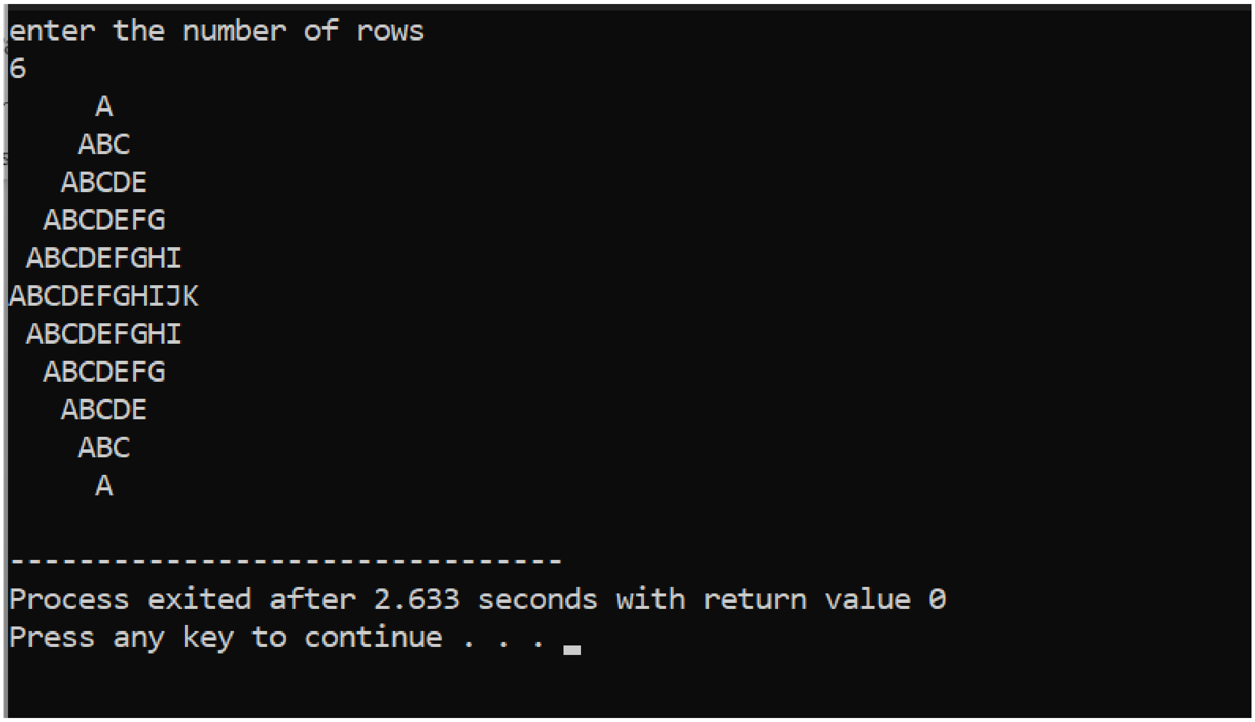
Input and Output:



```
Enter the Number of Rows: 5
    A
   ABC
  ABCDE
 ABCDEFG
ABCDEFGHI
 ABCDEFG
  ABCDE
   ABC
    A
```



```
//task #2 by pari batra 24k-3115
#include <iostream>
using namespace std;

int main() {
    int row,j;
    cout<<"enter the number of rows"<<endl;
    cin>>row;

    int i, space;

    for(i=1;i<=row;i++){
    for(space = 1; space <= row - i; space++){

            cout<<" ";
        }

        for( j = 1; j <= 2*i-1; j++) {
            cout << char('A' + j - 1);
        }
        cout << endl;
    }

     for(int i = row-1; i >= 1; i--) {

        for(int space = 1; space <= row - i; space++) {
            cout << " ";
        }


        for(int j = 1; j <= 2*i-1; j++) {
            cout << char('A' + j - 1);
        }
        cout << endl;
    }

    return 0;
```



```
enter the number of rows
6
     A
    ABC
   ABCDE
  ABCDEFG
 ABCDEFGHI
ABCDEFGHIJK
 ABCDEFGHI
  ABCDEFG
   ABCDE
    ABC
     A

--------------------------------
Process exited after 2.633 seconds with return value 0
Press any key to continue . . .
```

Task 3TASK # 03

Write a C++ program that defines a function named findPrimesInRange, which takes two numbers

as input and calculates all the prime numbers between them (exclusive). The program should

print all the prime numbers found within the given range.

Input:

Two integers, start and end, where the program will calculate and display all prime numbers

between start and end (excluding start and end).

Output:

A list of prime numbers between the given start and end.

Example:

Input:

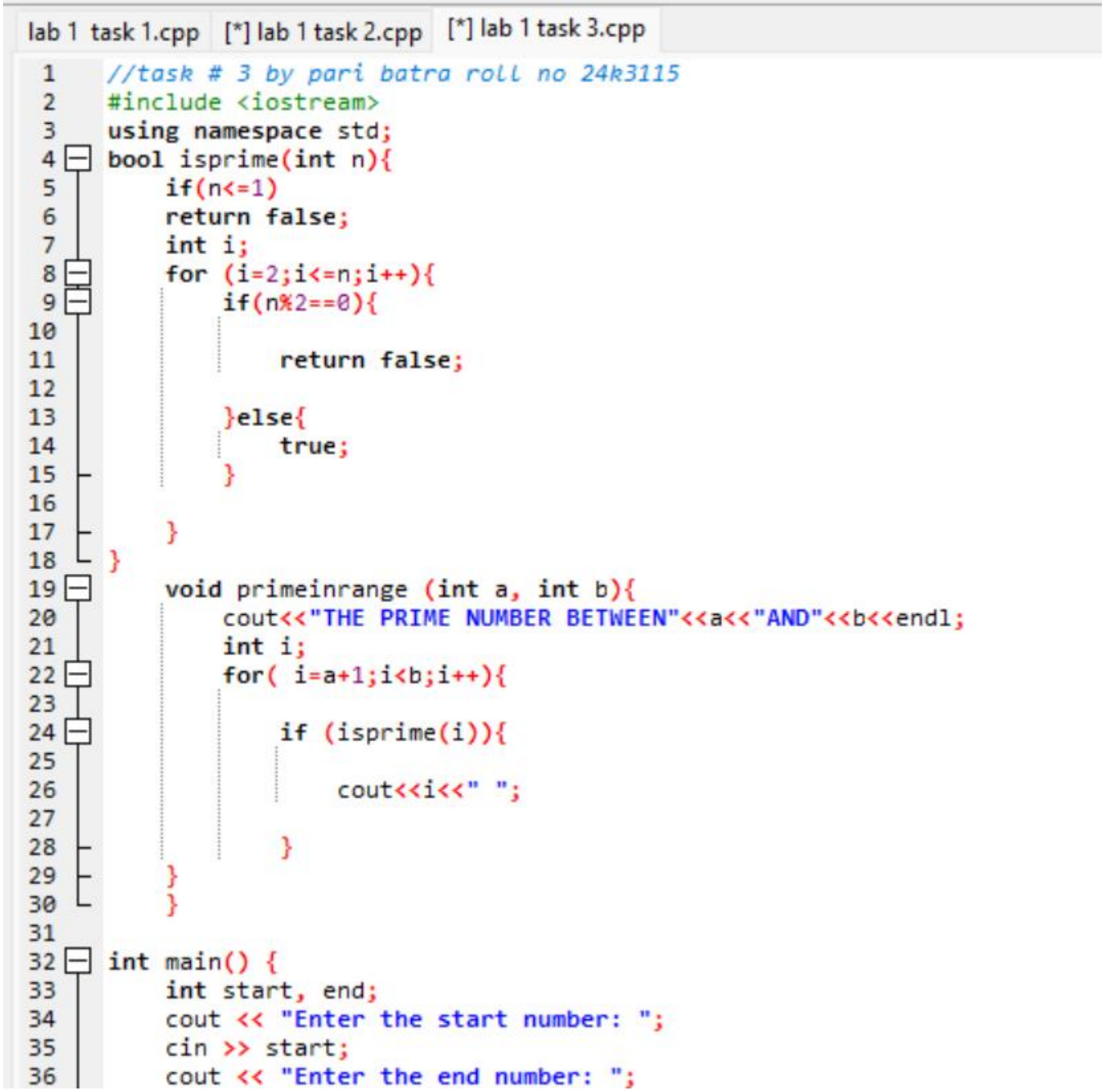
Enter the start number: 10

Enter the end numbe50

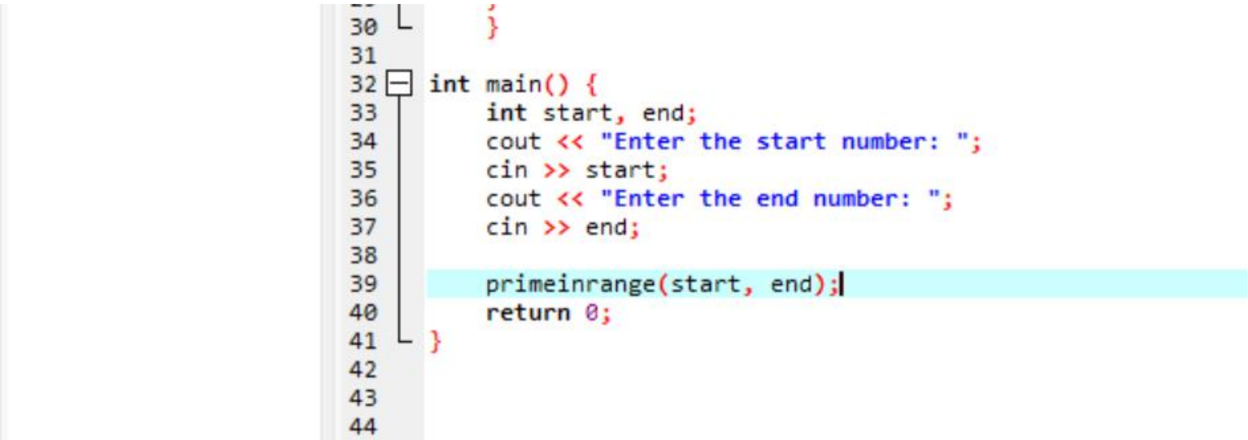
Output:

Prime numbers between 10 and 50 are:

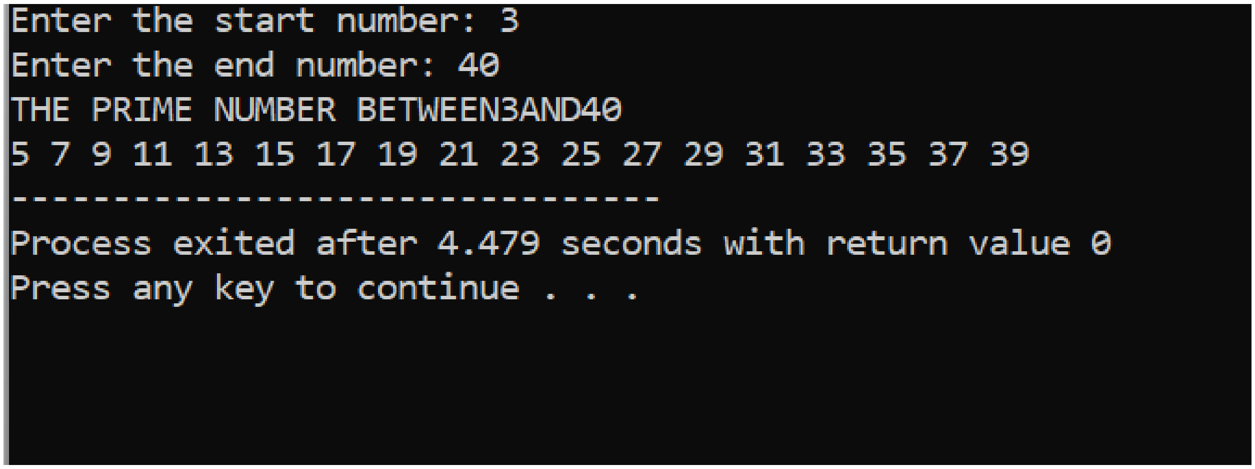
11 13 17 19 23 29 31 37 41 43 47



```cpp
//task # 3 by pari batra roll no 24k3115
#include <iostream>
using namespace std;
bool isprime(int n){
    if(n<=1)
    return false;
    int i;
    for (i=2;i<=n;i++){
        if(n%2==0){

            return false;

        }else{
            true;
        }

    }
}
    void primeinrange (int a, int b){
        cout<<"THE PRIME NUMBER BETWEEN"<<a<<"AND"<<b<<endl;
        int i;
        for( i=a+1;i<b;i++){

            if (isprime(i)){

                cout<<i<<" ";

            }
    }
    }

int main() {
    int start, end;
    cout << "Enter the start number: ";
    cin >> start;
    cout << "Enter the end number: ";
```



```
    }

int main() {
    int start, end;
    cout << "Enter the start number: ";
    cin >> start;
    cout << "Enter the end number: ";
    cin >> end;

    primeinrange(start, end);
    return 0;
}
```



```
Enter the start number: 3
Enter the end number: 40
THE PRIME NUMBER BETWEEN3AND40
5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
--------------------------------
Process exited after 4.479 seconds with return value 0
Press any key to continue . . .
```

Task 4

Write a C++ program that performs the following tasks:

1. Asks the user to input size and values for two arrays.
2. Swaps the values of both arrays.
3. Finds the largest element from the first array (after swapping).
4. Finds the second largest element from the second array (after swapping).
5. Finds the unique element (element that appears only once) in the first array (after swapping) and if all values in the array are unique, return the value at index 0.

Input:

- Two arrays of integers (1D) provided by the user. The program ensures that both arrays have the same size.

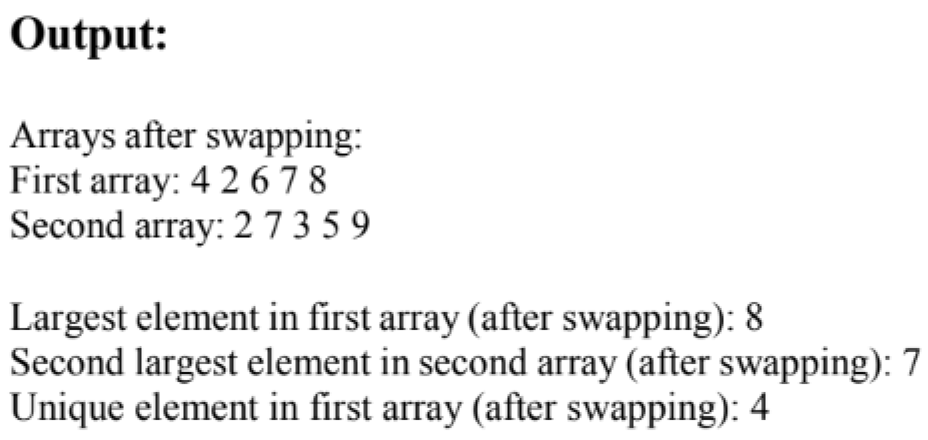
Output:

- The new swapped arrays.
- The largest element from the first swapped array.
- The second largest element from the second swapped array.
- The unique element in the first swapped array.

Example:

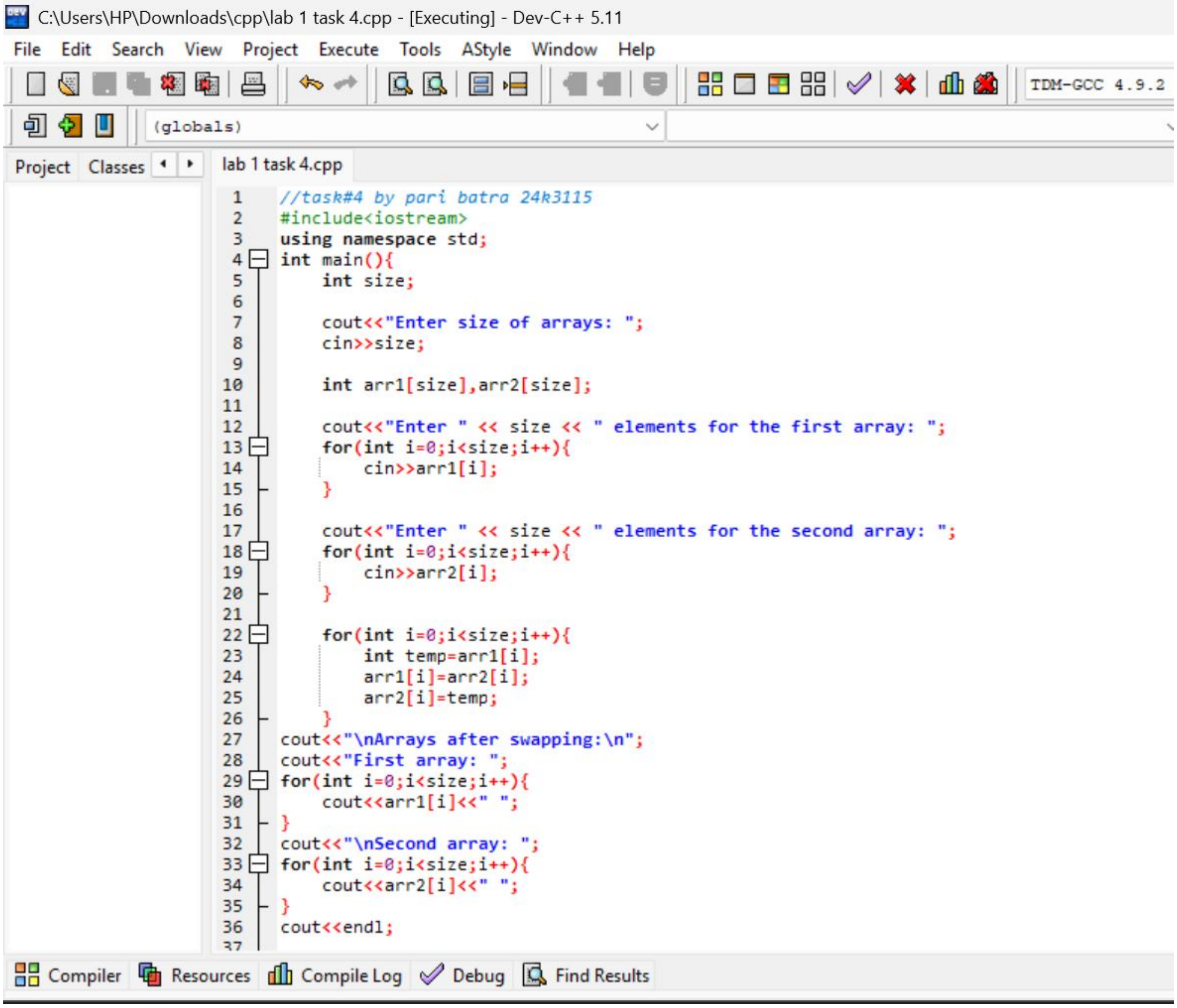
Input:

```
Enter size of arrays: 5
Enter 5 elements for the first array: 2 7 3 5 9
Enter 5 elements for the second array: 4 2 6 7 8
```



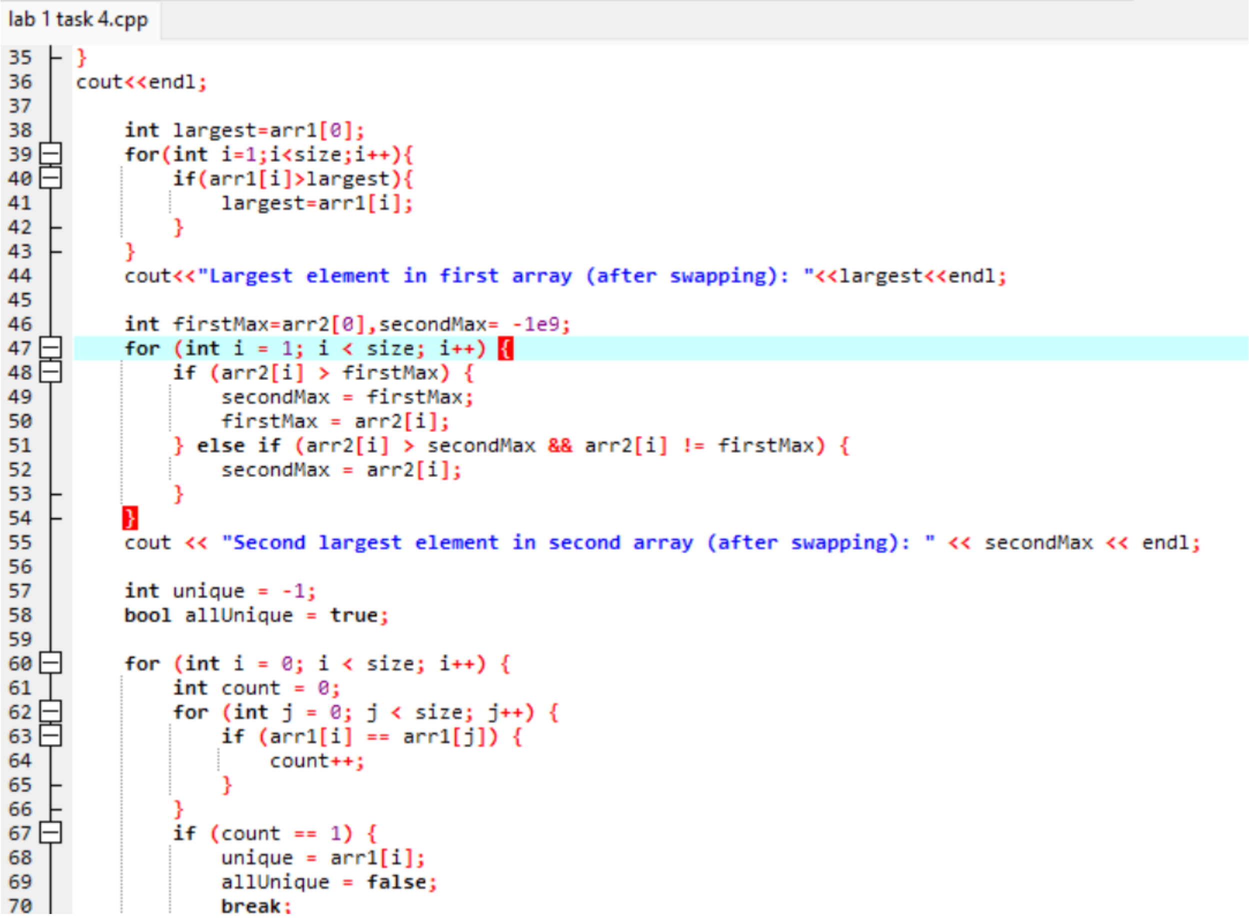
Output:

```
Arrays after swapping:
First array: 4 2 6 7 8
Second array: 2 7 3 5 9

Largest element in first array (after swapping): 8
Second largest element in second array (after swapping): 7
Unique element in first array (after swapping): 4
```



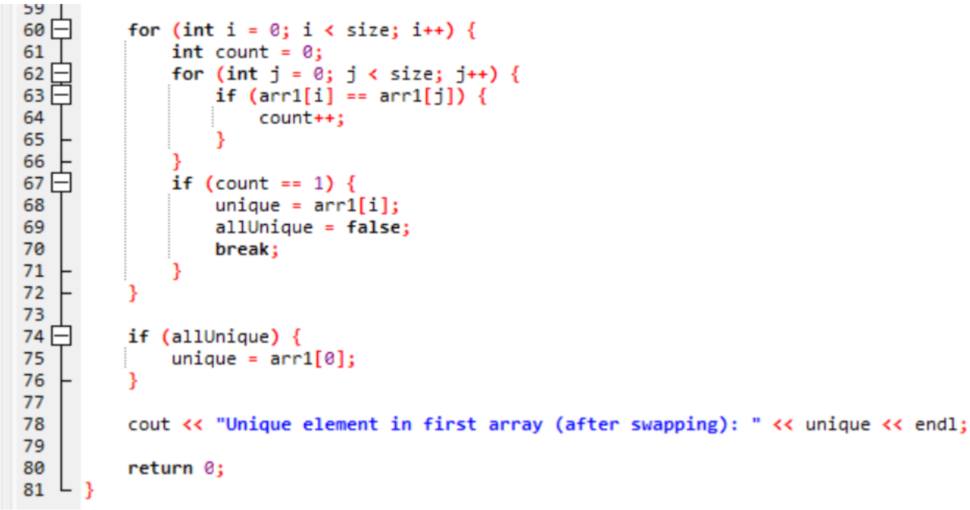
C:\Users\HP\Downloads\cpp\lab 1 task 4.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2
(globals)
Project Classes
lab 1 task 4.cpp
Compiler Resources Compile Log Debug Find Results

```cpp
//task#4 by pari batra 24k3115
#include<iostream>
using namespace std;
int main(){
    int size;

    cout<<"Enter size of arrays: ";
    cin>>size;

    int arr1[size],arr2[size];

    cout<<"Enter " << size << " elements for the first array: ";
    for(int i=0;i<size;i++){
        cin>>arr1[i];
    }

    cout<<"Enter " << size << " elements for the second array: ";
    for(int i=0;i<size;i++){
        cin>>arr2[i];
    }

    for(int i=0;i<size;i++){
        int temp=arr1[i];
        arr1[i]=arr2[i];
        arr2[i]=temp;
    }
cout<<"\nArrays after swapping:\n";
cout<<"First array: ";
for(int i=0;i<size;i++){
    cout<<arr1[i]<<" ";
}
cout<<"\nSecond array: ";
for(int i=0;i<size;i++){
    cout<<arr2[i]<<" ";
}
cout<<endl;
```

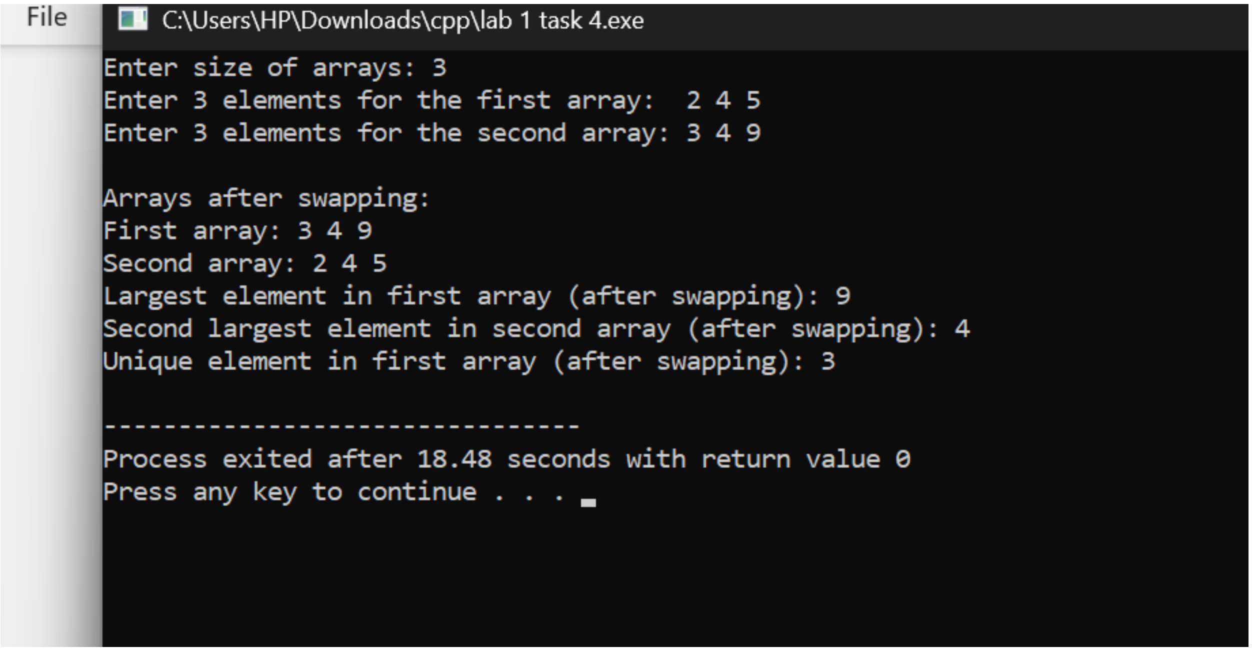


lab 1 task 4.cpp

```cpp
}
cout<<endl;

    int largest=arr1[0];
    for(int i=1;i<size;i++){
        if(arr1[i]>largest){
            largest=arr1[i];
        }
    }
    cout<<"Largest element in first array (after swapping): "<<largest<<endl;

    int firstMax=arr2[0],secondMax= -1e9;
    for (int i = 1; i < size; i++) {
        if (arr2[i] > firstMax) {
            secondMax = firstMax;
            firstMax = arr2[i];
        } else if (arr2[i] > secondMax && arr2[i] != firstMax) {
            secondMax = arr2[i];
        }
    }
    cout << "Second largest element in second array (after swapping): " << secondMax << endl;

    int unique = -1;
    bool allUnique = true;

    for (int i = 0; i < size; i++) {
        int count = 0;
        for (int j = 0; j < size; j++) {
            if (arr1[i] == arr1[j]) {
                count++;
            }
        }
        if (count == 1) {
            unique = arr1[i];
            allUnique = false;
            break;
```



```cpp

    for (int i = 0; i < size; i++) {
        int count = 0;
        for (int j = 0; j < size; j++) {
            if (arr1[i] == arr1[j]) {
                count++;
            }
        }
        if (count == 1) {
            unique = arr1[i];
            allUnique = false;
            break;
        }
    }

    if (allUnique) {
        unique = arr1[0];
    }

    cout << "Unique element in first array (after swapping): " << unique << endl;

    return 0;
}
```



```
Enter size of arrays: 3
Enter 3 elements for the first array:  2 4 5
Enter 3 elements for the second array: 3 4 9

Arrays after swapping:
First array: 3 4 9
Second array: 2 4 5
Largest element in first array (after swapping): 9
Second largest element in second array (after swapping): 4
Unique element in first array (after swapping): 3

--------------------------------
Process exited after 18.48 seconds with return value 0
Press any key to continue . . .
```

Task 5

PROGRAMMING TASKS FOR LAB # 01

TASK # 05

Write a C++ program to multiply two matrices using a **function** named multiplyMatrices. The program should take two 2-dimensional arrays (matrices) as input, validate the multiplication rule, and calculate the product of the matrices. Use the following instructions:

Rules for Matrix Multiplication:

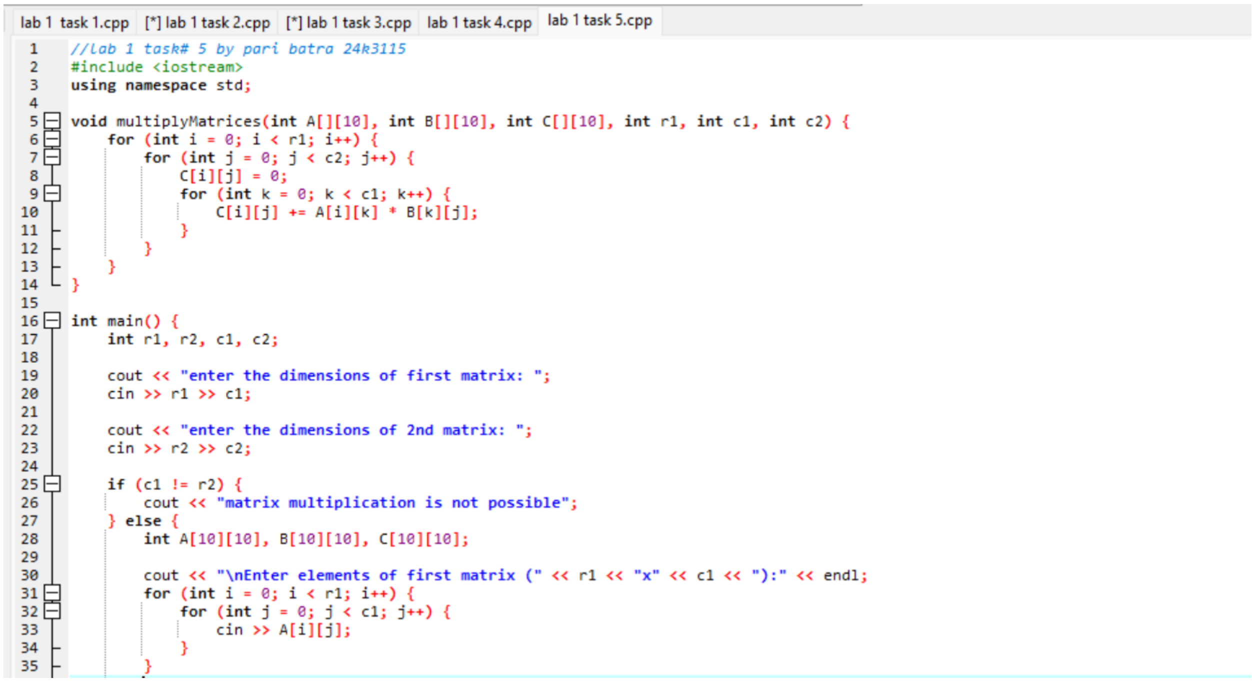
1. The **number of columns** in the first matrix must equal the **number of rows** in the second matrix.

Input:

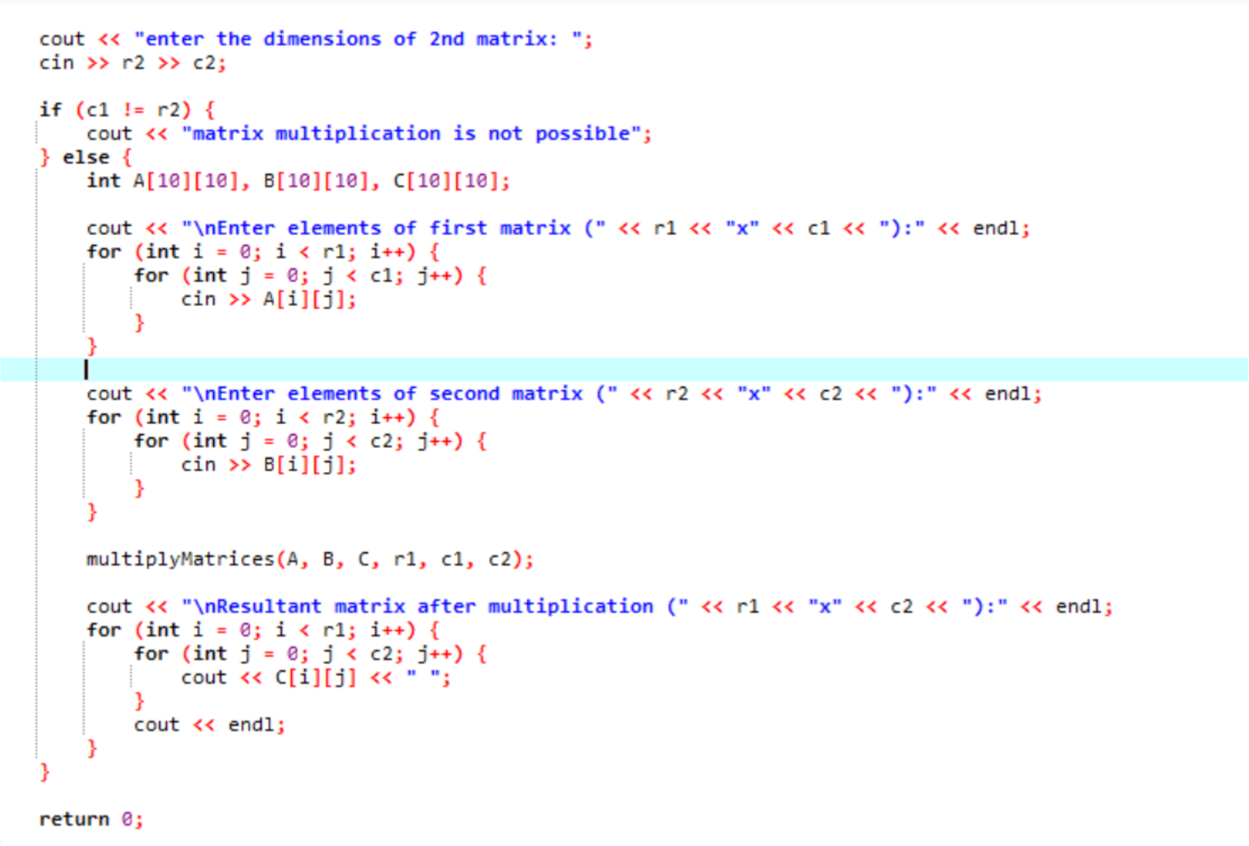
1. Dimensions of the first matrix **A** (m rows and n columns).
2. Dimensions of the second matrix **B** (n rows and p columns).
3. Elements of both matrices **A** and **B**.

Output:

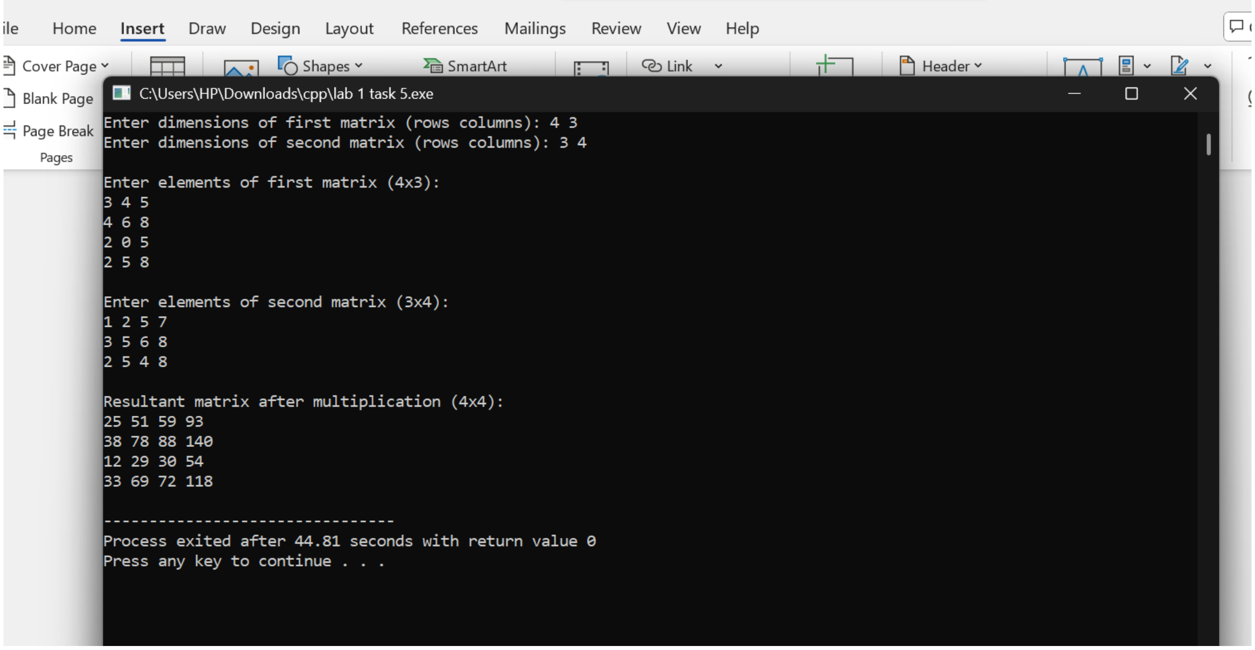
1. The resulting matrix **C** after multiplication.



```cpp
//lab 1 task# 5 by pari batra 24k3115
#include <iostream>
using namespace std;

void multiplyMatrices(int A[][10], int B[][10], int C[][10], int r1, int c1, int c2) {
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            C[i][j] = 0;
            for (int k = 0; k < c1; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }
}

int main() {
    int r1, r2, c1, c2;

    cout << "enter the dimensions of first matrix: ";
    cin >> r1 >> c1;

    cout << "enter the dimensions of 2nd matrix: ";
    cin >> r2 >> c2;

    if (c1 != r2) {
        cout << "matrix multiplication is not possible";
    } else {
        int A[10][10], B[10][10], C[10][10];

        cout << "\nEnter elements of first matrix (" << r1 << "x" << c1 << "):" << endl;
        for (int i = 0; i < r1; i++) {
            for (int j = 0; j < c1; j++) {
                cin >> A[i][j];
            }
        }
```



```cpp
cout << "enter the dimensions of 2nd matrix: ";
cin >> r2 >> c2;

if (c1 != r2) {
    cout << "matrix multiplication is not possible";
} else {
    int A[10][10], B[10][10], C[10][10];

    cout << "\nEnter elements of first matrix (" << r1 << "x" << c1 << "):" << endl;
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c1; j++) {
            cin >> A[i][j];
        }
    }

    cout << "\nEnter elements of second matrix (" << r2 << "x" << c2 << "):" << endl;
    for (int i = 0; i < r2; i++) {
        for (int j = 0; j < c2; j++) {
            cin >> B[i][j];
        }
    }

    multiplyMatrices(A, B, C, r1, c1, c2);

    cout << "\nResultant matrix after multiplication (" << r1 << "x" << c2 << "):" << endl;
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            cout << C[i][j] << " ";
        }
        cout << endl;
    }
}

return 0;
```



```
Enter dimensions of first matrix (rows columns): 4 3
Enter dimensions of second matrix (rows columns): 3 4

Enter elements of first matrix (4x3):
3 4 5
4 6 8
2 0 5
2 5 8

Enter elements of second matrix (3x4):
1 2 5 7
3 5 6 8
2 5 4 8

Resultant matrix after multiplication (4x4):
25 51 59 93
38 78 88 140
12 29 30 54
33 69 72 118

--------------------------------
Process exited after 44.81 seconds with return value 0
Press any key to continue . . .
```