**Name:**

Ayesha jami , sarah Iftikhar , Shaheera , Fatima Ejaz

**1. Linear Search**

**Description:**

* **Linear Search** involves checking each element of the list or array sequentially, one by one, from the beginning to the end until the target element is found or the entire list is searched.
* It is used for both **sorted** and **unsorted** arrays.

**Steps:**

1. Start at the first element.
2. Compare the element with the target.
3. If they match, return the index of the element.
4. If no match is found, move to the next element.
5. Continue until the target is found or all elements are checked.

**2. Binary Search**

**Description:**

* **Binary Search** works by repeatedly dividing the sorted array in half and comparing the target value with the middle element. Based on the comparison, it either searches the left half or the right half, effectively reducing the search space by half each time.
* **Binary Search** requires the array to be **sorted** beforehand.

**Steps:**

1. Start with the entire sorted array.
2. Compare the middle element with the target.
3. If they match, return the middle index.
4. If the target is smaller than the middle element, repeat the search in the left half.
5. If the target is larger, repeat the search in the right half.
6. Continue dividing the search space in half until the target is found or the search space is empty.

**Program 3:**

**Write a program to read 10 integers from the user and store them in a single-dimension array. Then, print the array in reverse order.**

#include <iostream>

using namespace std;

int main()

{

int arr[10];

cout << "Enter 10 integers:" << endl;

for (int i = 0; i < 10; i++) {

cin >> arr[i];

}

cout << "Array in reverse order:" << endl;

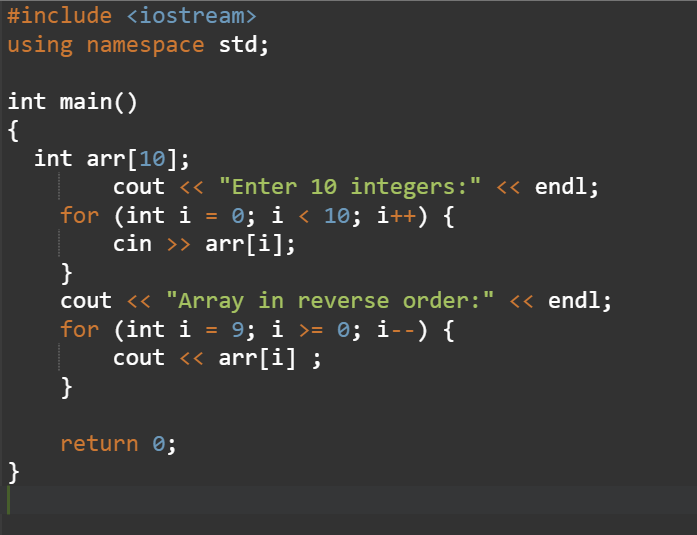
for (int i = 9; i >= 0; i--) {

cout << arr[i] ;

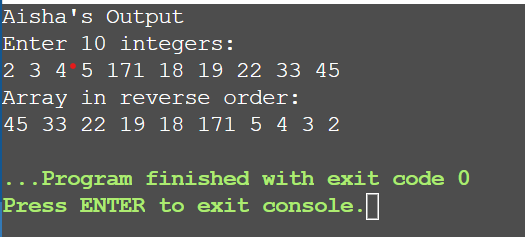
}

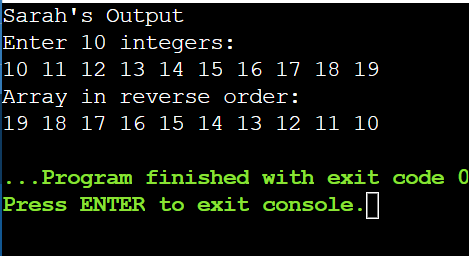
return 0;

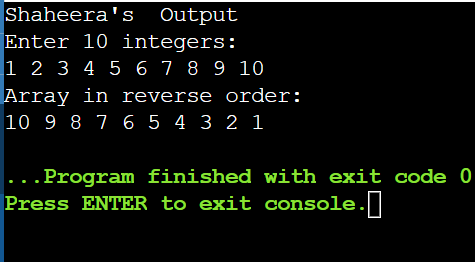
}

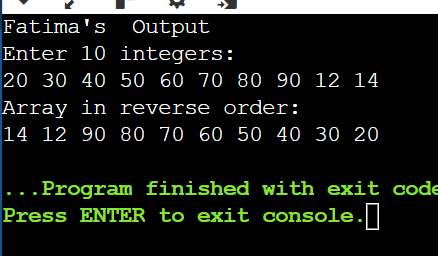


**Output:**

****







**Program 4:**

**Write a program to take a string as input from the user and check whether the string is a palindrome using a null-terminated character array.**

#include <iostream>

#include <string>

#include <algorithm>

using namespace std;

int main()

{

string str;

cout << "Enter a string: ";

cin >> str;

string reversedstr = str;

reverse(reversedstr.begin(), reversedstr.end());

if (str == reversedstr)

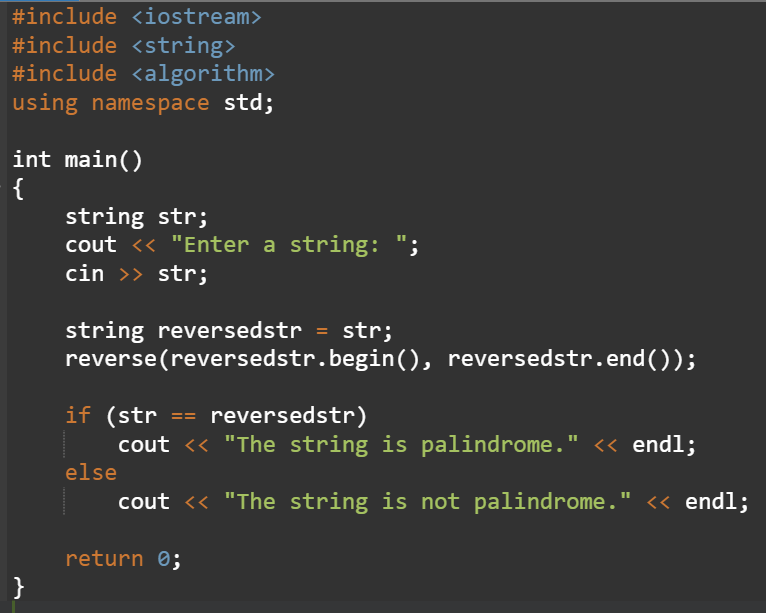
cout << "The string is palindrome." << endl;

else

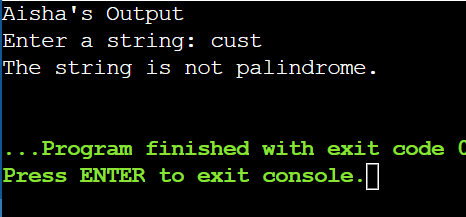
cout << "The string is not palindrome." << endl;

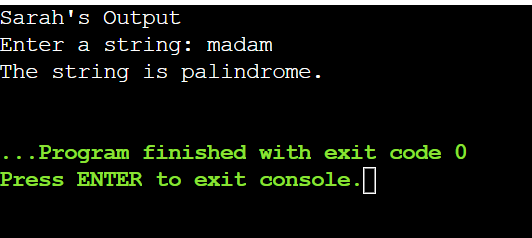
return 0;

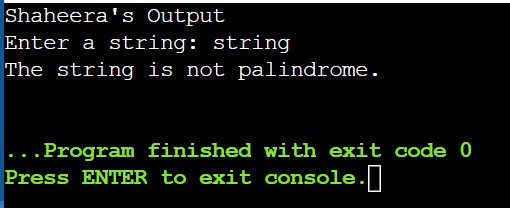
}

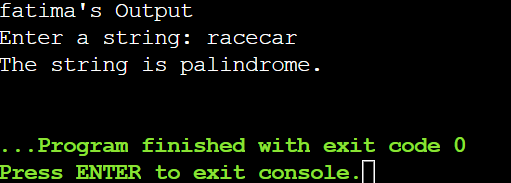


**Output:**









**Program 5:**

**Write a program to sort an array of integers in ascending order using the bubble sort algorithm.**

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter the number of elements: ";

cin >> n;

int arr[n];

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - 1 - i; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

cout << "Sorted array in ascending order: ";

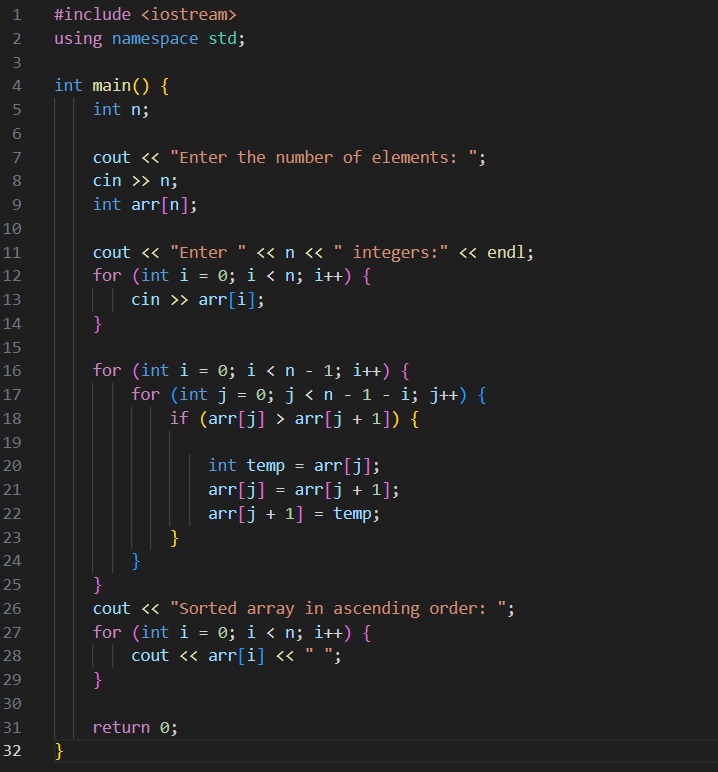
for (int i = 0; i < n; i++) {

cout << arr[i] << " ";

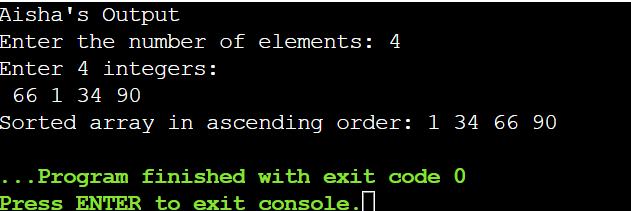
}

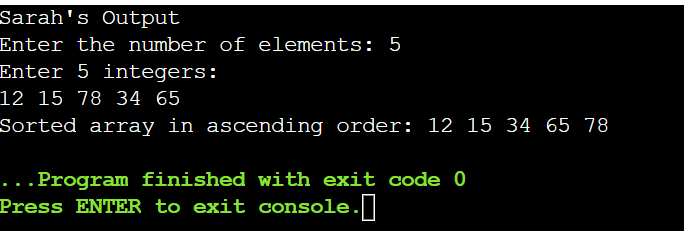
return 0;

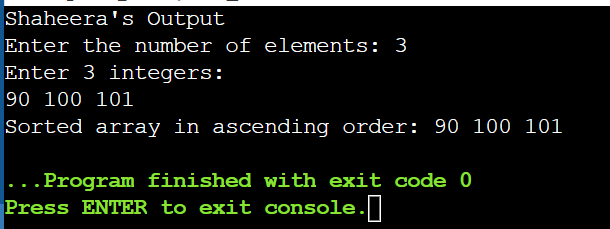
}

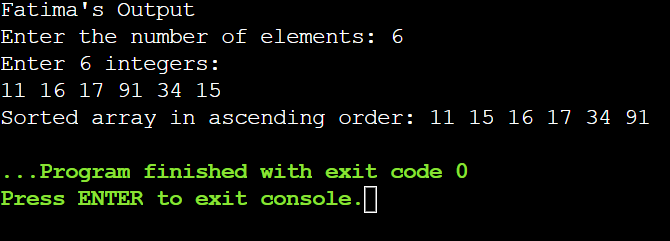


**Output:**









**Program 6:**

**Write a program to search for a specific element in an array and print the index of the element if found, otherwise print a message that the element is not in the array.**

#include <iostream>

using namespace std;

int main() {

int n, element, index = -1;

cout << "Enter the number of elements: ";

cin >> n;

int arr[n];

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

cout << "Enter the element to search for: ";

cin >> element;

for (int i = 0; i < n; i++) {

if (arr[i] == element) {

index = i;

break;

}

}

if (index != -1) {

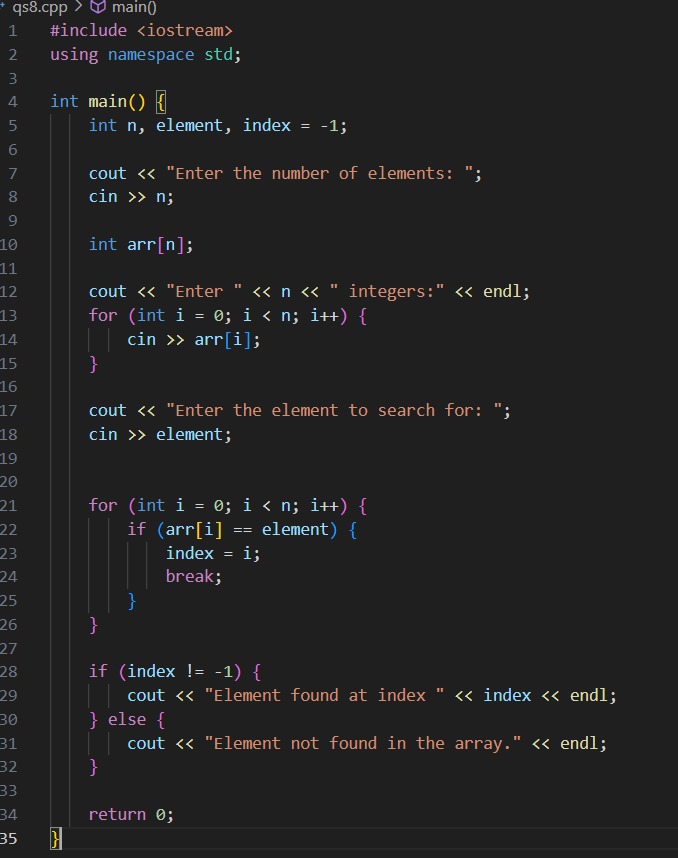
cout << "Element found at index " << index << endl;

} else {

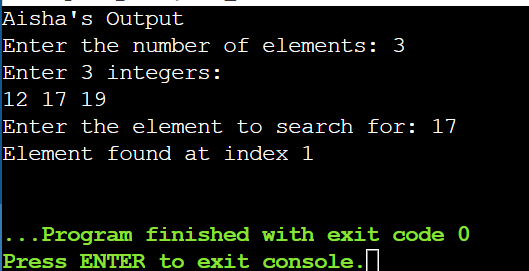
cout << "Element not found in the array." << endl;

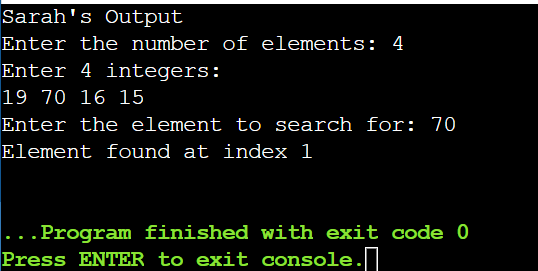
}

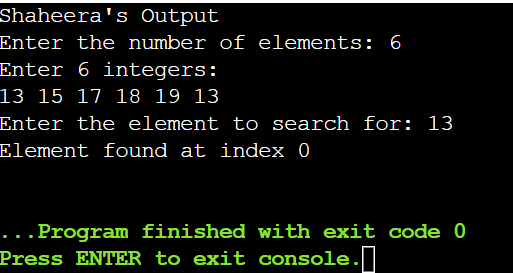
return 0;

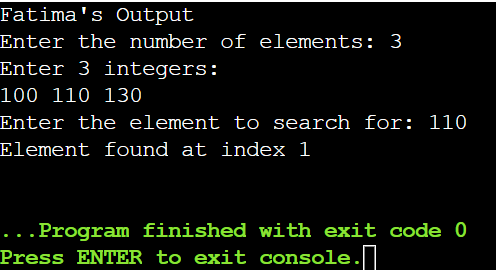
}

**Output:**









**program7:**

**Write a program to find the minimum and maximum values from an array of integers and print both values**

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter the number of elements: ";

cin >> n;

int arr[n];

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; i++) {

cin >> arr[i];

}

int minValue = arr[0];

int maxValue = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] < minValue) {

minValue = arr[i];

}

if (arr[i] > maxValue) {

maxValue = arr[i];

}

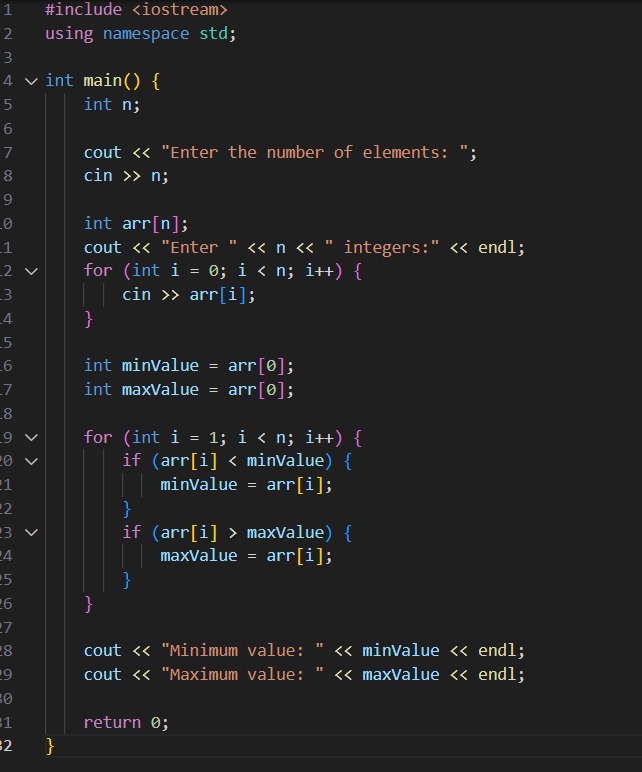
}

cout << "Minimum value: " << minValue << endl;

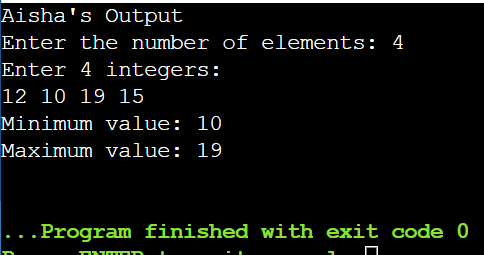
cout << "Maximum value: " << maxValue << endl;

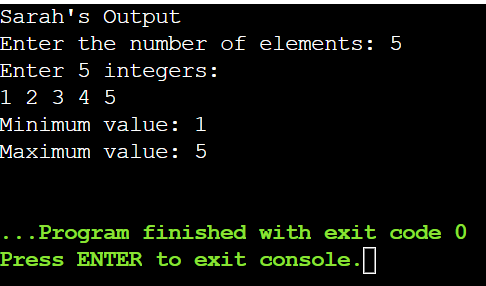
return 0;

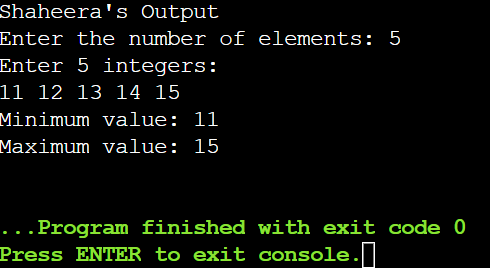
}

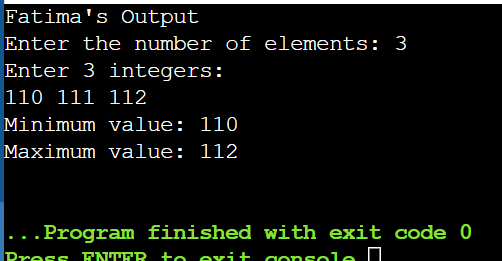


**Output:**









**program8:**

**Write a program to store and print a 2x3 matrix (two rows, three columns) of integers.**

#include <iostream>

using namespace std;

int main() {

int matrix[2][3];

cout << "Enter the elements for a 2x3 matrix:" << endl;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

cout << "Element at position [" << i << "][" << j << "]: ";

cin >> matrix[i][j];

}

}

cout << "\nThe 2x3 matrix is:" << endl;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 3; j++) {

cout << matrix[i][j] << " ";

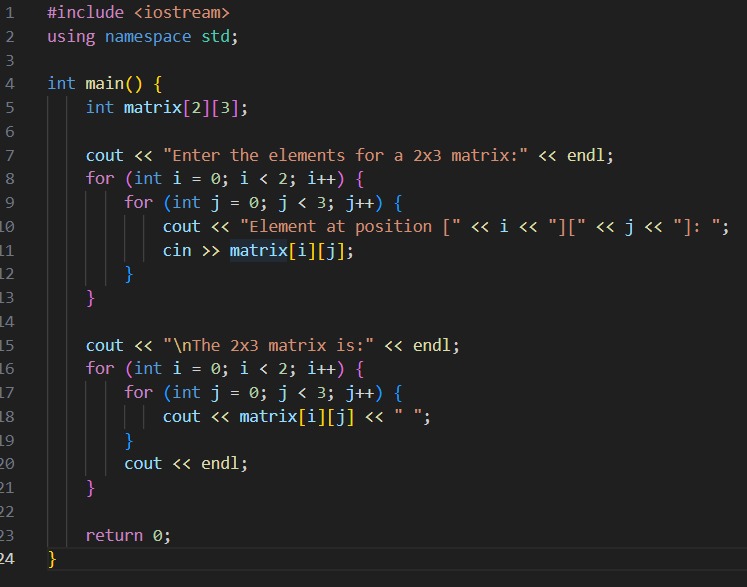
}

cout << endl;

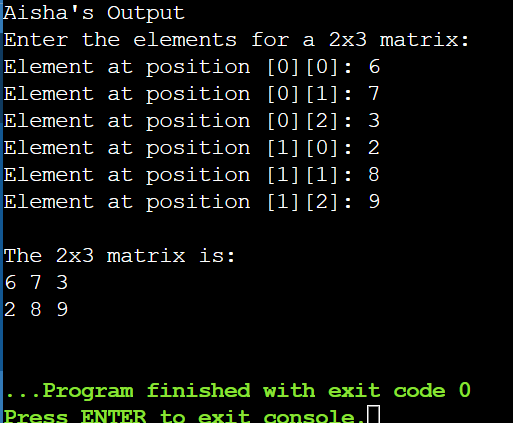
}

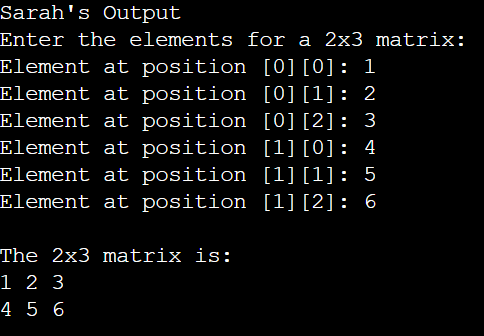
return 0;

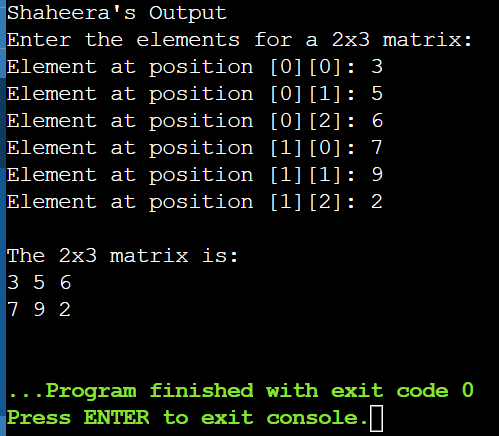
}

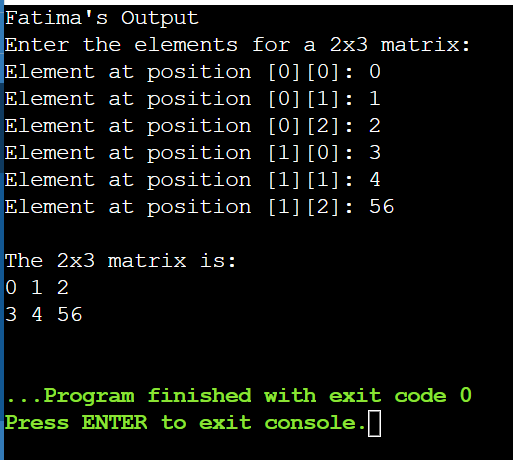


**Output:**









**Program 9:**

**Write a program to find the sum of all elements in a 3x3 matrix (three rows, three columns) of integers**.

#include <iostream>

using namespace std;

int main() {

int matrix[3][3];

int sum = 0;

cout << "Enter 9 elements for a 3x3 matrix:" << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cout << "Enter element at position (" << i+1 << "," << j+1 << "): ";

cin >> matrix[i][j];

}

}

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

sum += matrix[i][j];

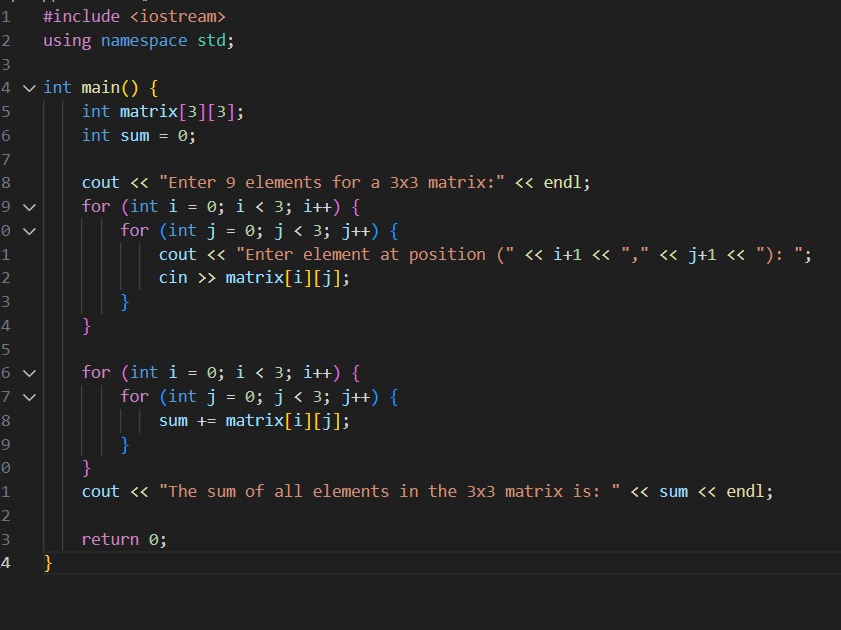
}

}

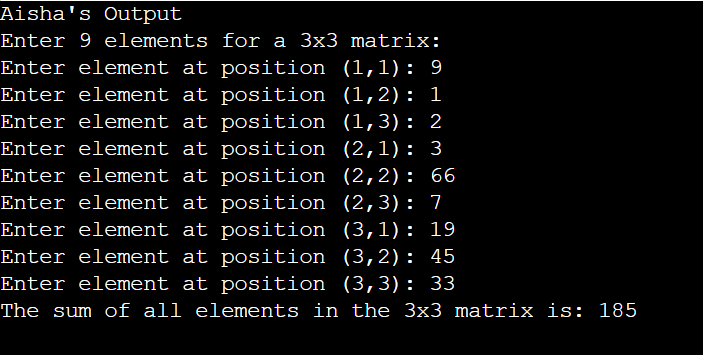
cout << "The sum of all elements in the 3x3 matrix is: " << sum << endl;

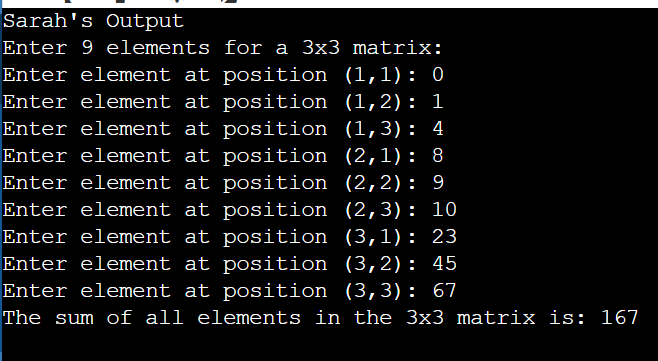
return 0;

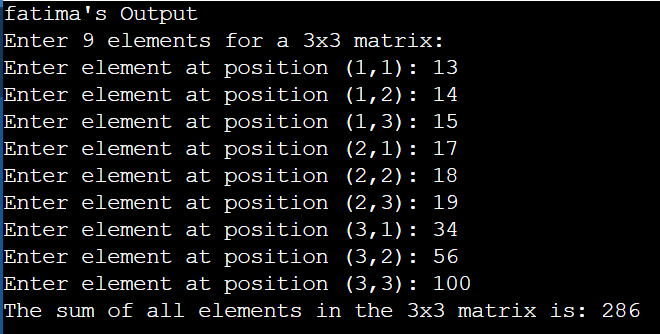
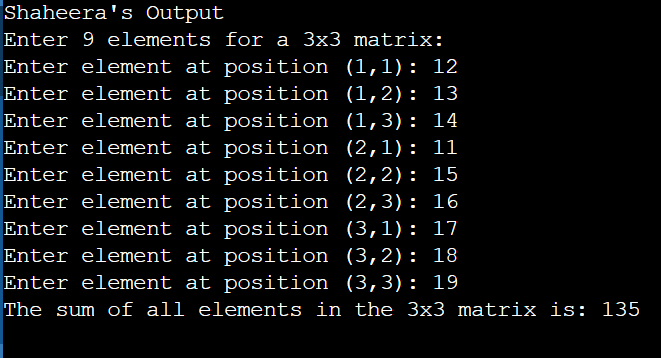
}



**Output:**







**Program10:**

**Write a program to multiply two 2x2 matrices and print the resulting matrix.**

#include <iostream>

using namespace std;

int main() {

int matrix1[2][2], matrix2[2][2], result[2][2];

cout << "Enter elements for the first 2x2 matrix:" << endl;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

cout << "Enter element at position [" << i+1 << "][" << j+1 << "]: ";

cin >> matrix1[i][j];

}

}

cout << "Enter elements for the second 2x2 matrix:" << endl;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

cout << "Enter element at position [" << i+1 << "][" << j+1 << "]: ";

cin >> matrix2[i][j];

}

}

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

result[i][j] = 0;

}

}

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

for (int k = 0; k < 2; k++) {

result[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

cout << "\nThe resulting matrix after multiplication is:" << endl;

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

cout << result[i][j] << " ";

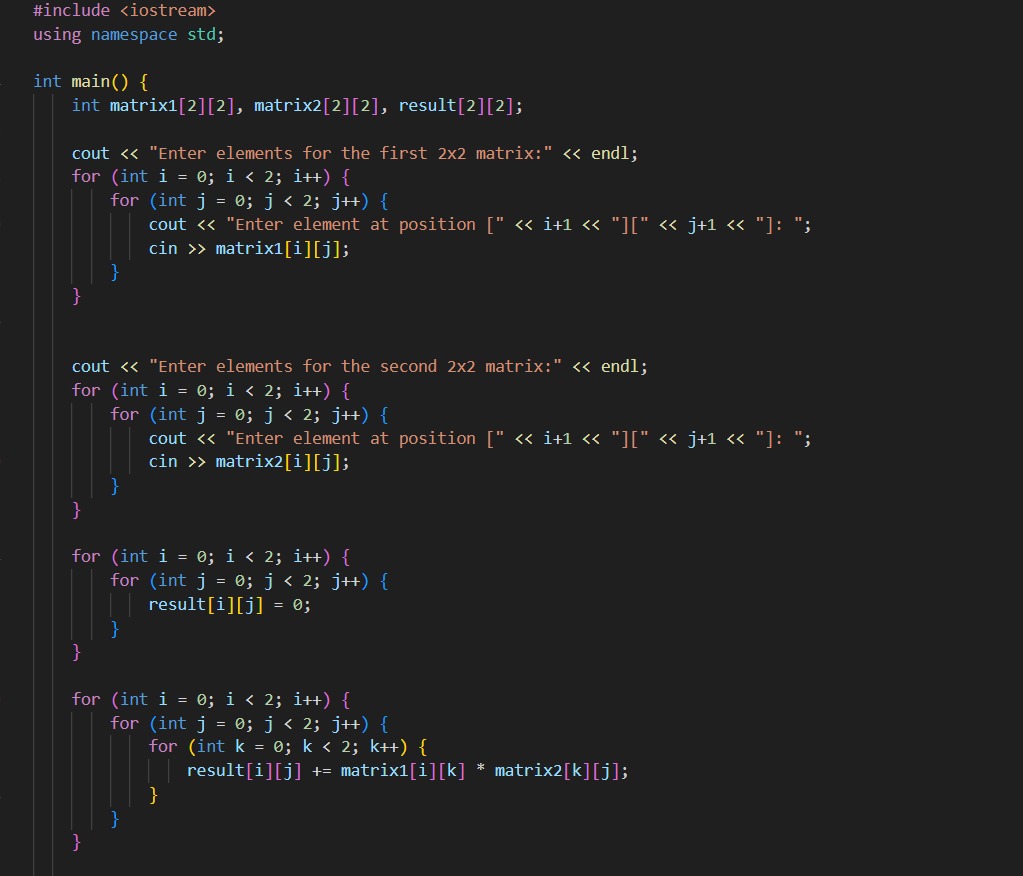
}

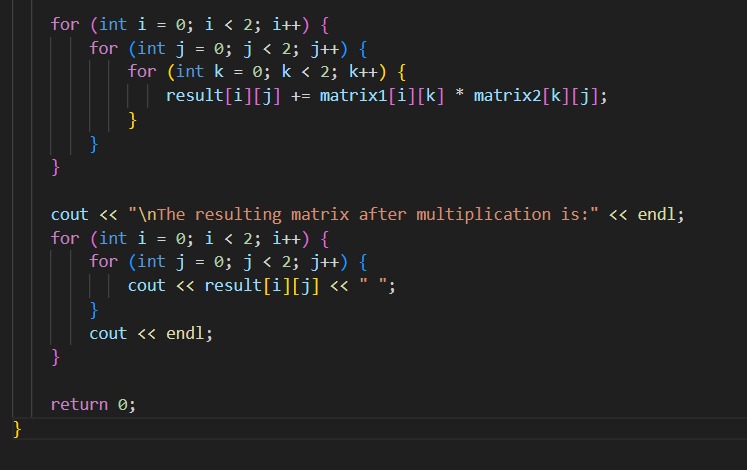
cout << endl;

}

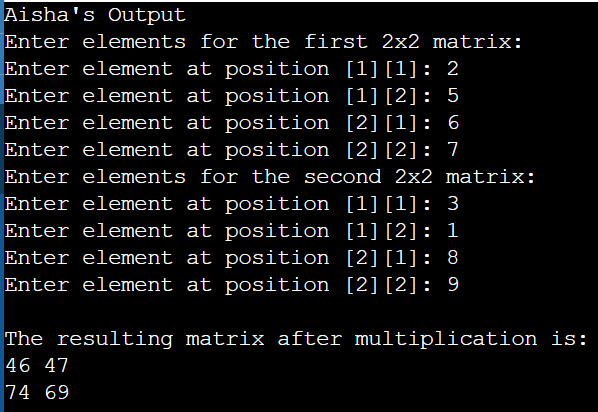
return 0;

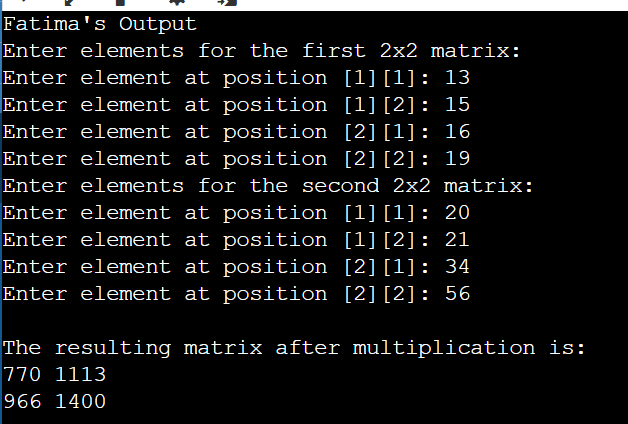
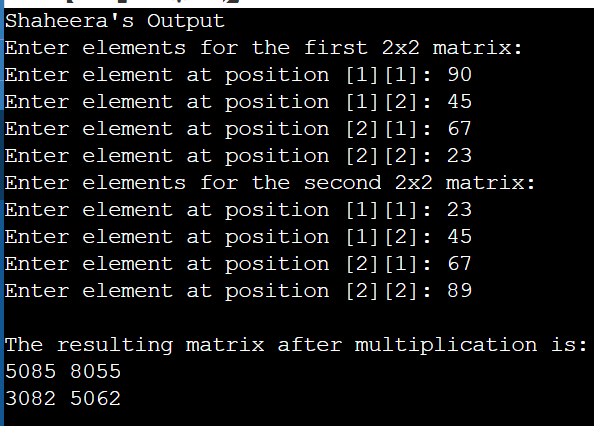
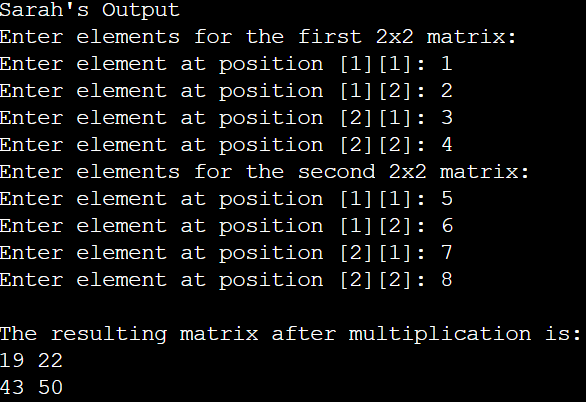
}





**Output:**





**Program 11:**

**Write a program to find the largest element in a 2D array.**

#include <iostream>

using namespace std;

int main() {

int rows = 3, cols = 3;

int matrix[3][3];

int largest;

cout << "Enter elements for a 3x3 matrix:" << endl;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

cout << "Enter element at position [" << i+1 << "][" << j+1 << "]: ";

cin >> matrix[i][j];

}

}

largest = matrix[0][0];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

if (matrix[i][j] > largest) {

largest = matrix[i][j];

}

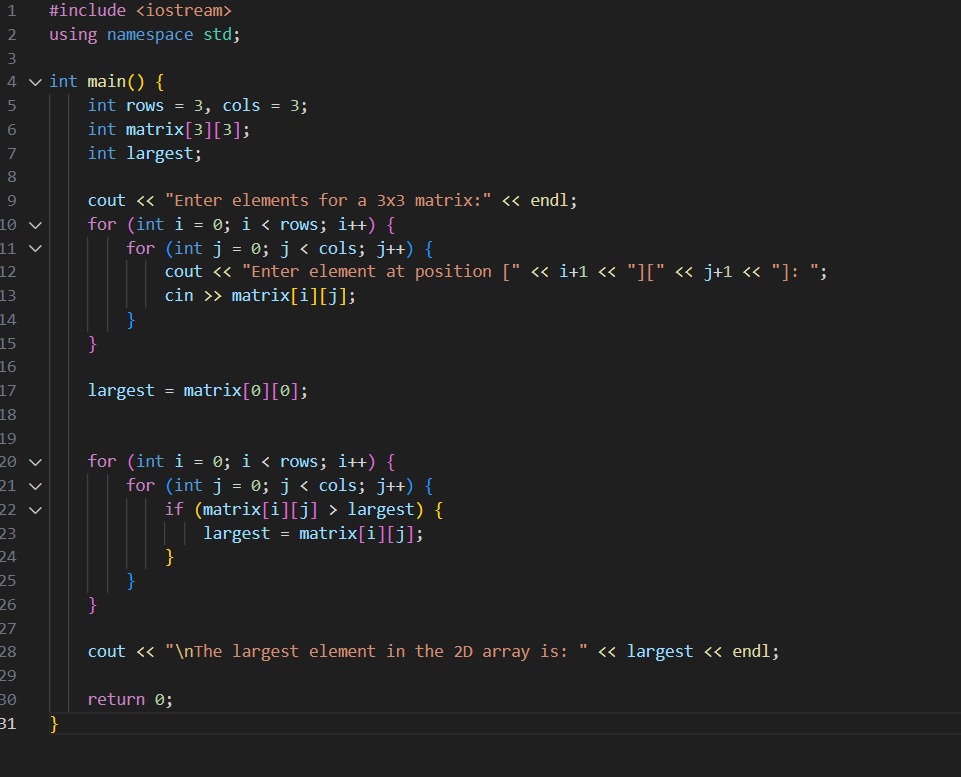
}

}

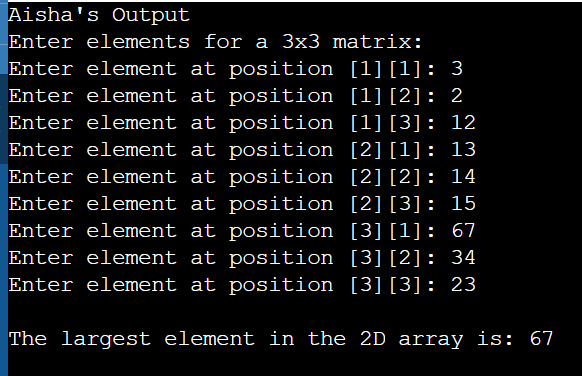
cout << "\nThe largest element in the 2D array is: " << largest << endl;

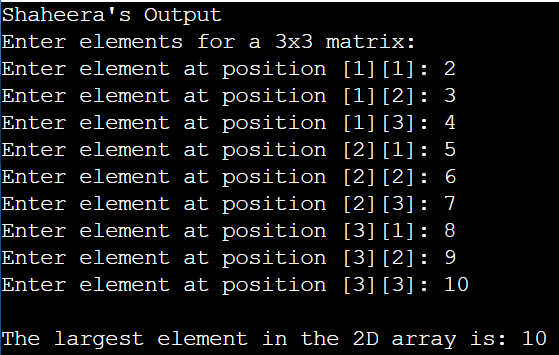
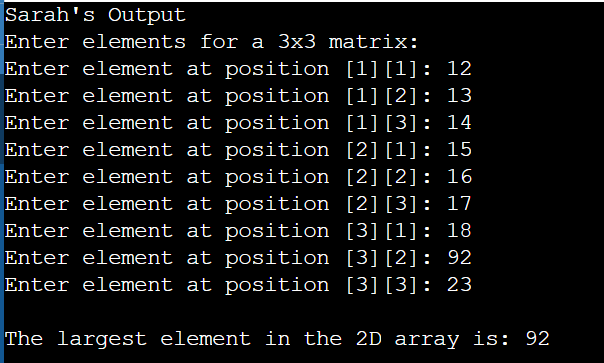
return 0;

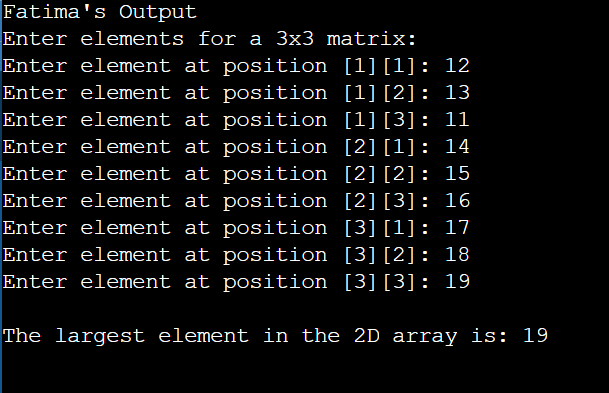
}



**Output:**







**Program 12**

**Write a program to calculate the average of elements in an array of integers and display the result.**

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Enter the number of elements in the array: ";

cin >> n;

int arr[n];

int sum = 0;

cout << "Enter " << n << " elements:" << endl;

for (int i = 0; i < n; i++) {

cout << "Element " << i + 1 << ": ";

cin >> arr[i];

sum += arr[i];

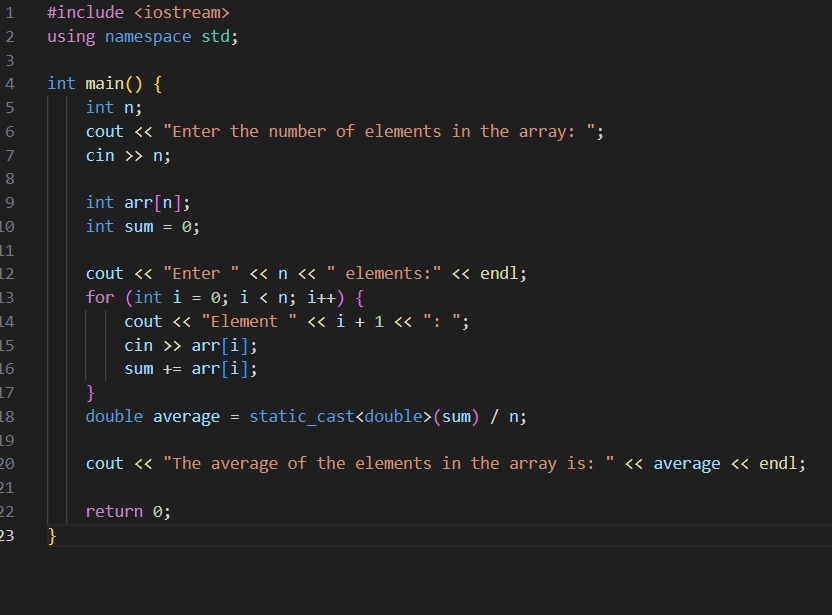
}

double average = static\_cast<double>(sum) / n;

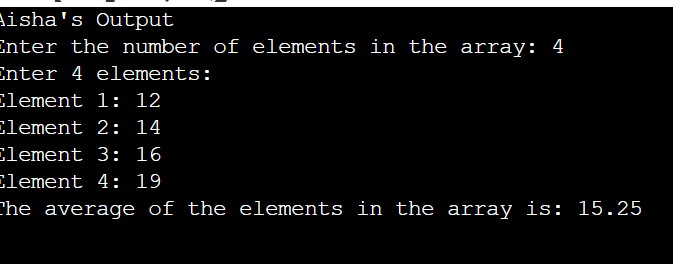
cout << "The average of the elements in the array is: " << average << endl;

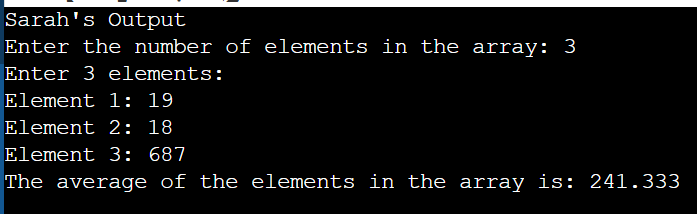
return 0;

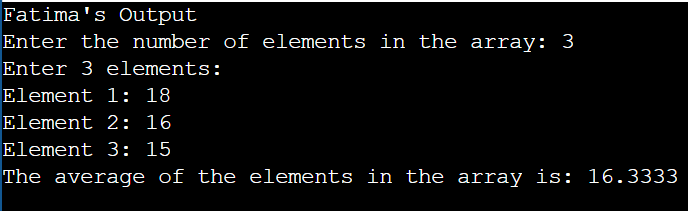
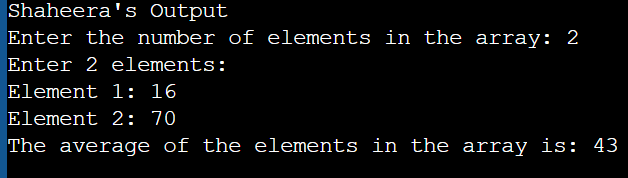
}



**Output:**







**Program 13**

**Write a program to transpose a 3x3 matrix (swap rows and columns).**

#include <iostream>

using namespace std;

int main() {

int matrix[3][3], transpose[3][3];

cout << "Enter elements for a 3x3 matrix:" << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cout << "Enter element at position [" << i + 1 << "][" << j + 1 << "]: ";

cin >> matrix[i][j];

}

}

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

transpose[j][i] = matrix[i][j];

}

}

cout << "\nThe transposed matrix is:" << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cout << transpose[i][j] << " ";

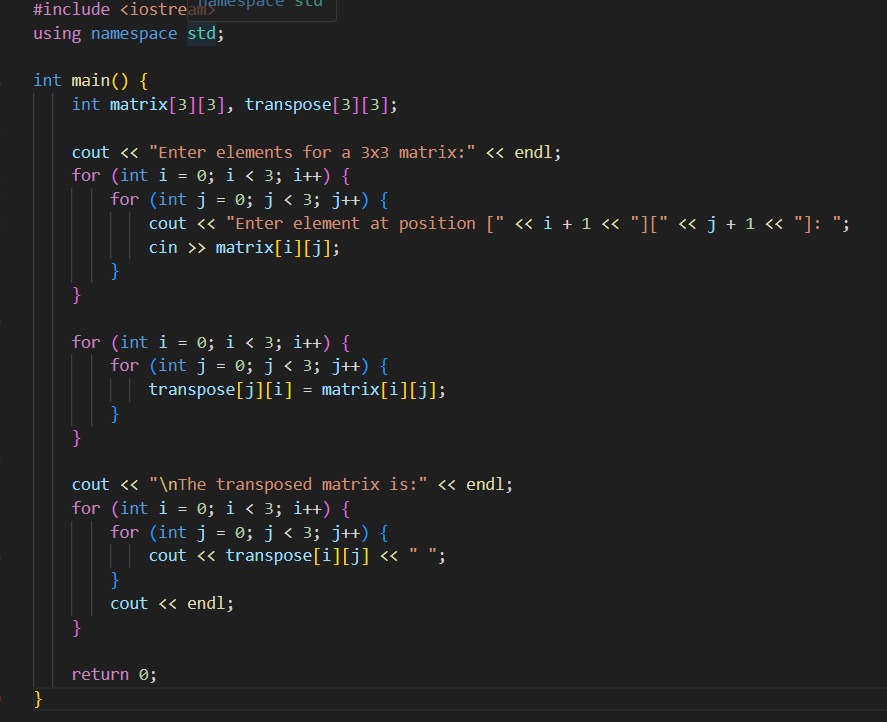
}

cout << endl;

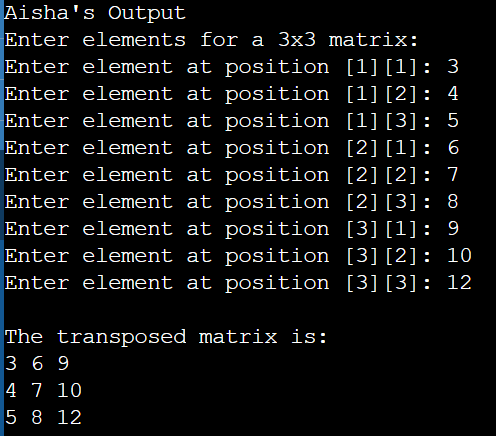
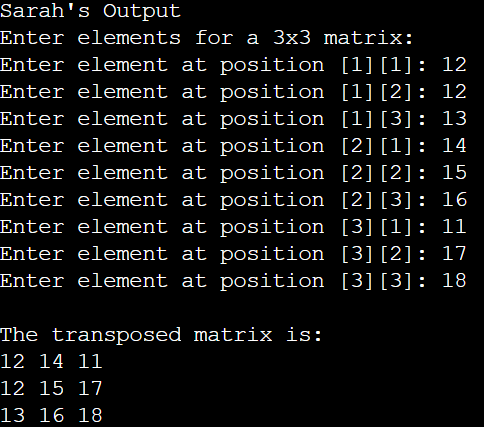
}

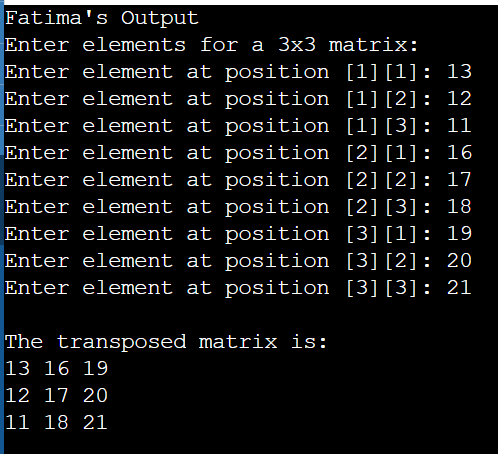
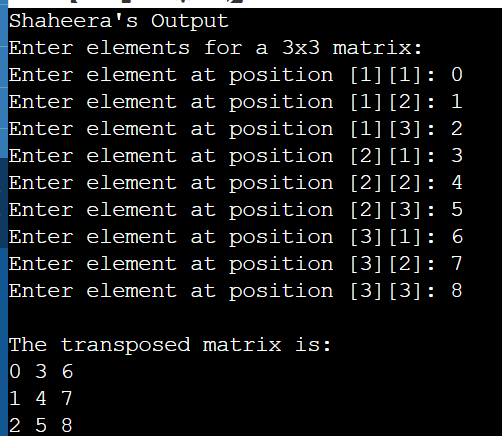
return 0;

}



**Output:**

****



**Program 14:**

**Write a program to calculate the sum of both diagonals in a 3x3 matrix.**

#include <iostream>

using namespace std;

int main() {

int matrix[3][3];

int primaryDiagonalSum = 0, secondaryDiagonalSum = 0;

cout << "Enter elements for a 3x3 matrix:" << endl;

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

cout << "Enter element at position [" << i + 1 << "][" << j + 1 << "]: ";

cin >> matrix[i][j];

}

}

for (int i = 0; i < 3; i++) {

primaryDiagonalSum += matrix[i][i];

}

for (int i = 0; i < 3; i++) {

secondaryDiagonalSum += matrix[i][2 - i];

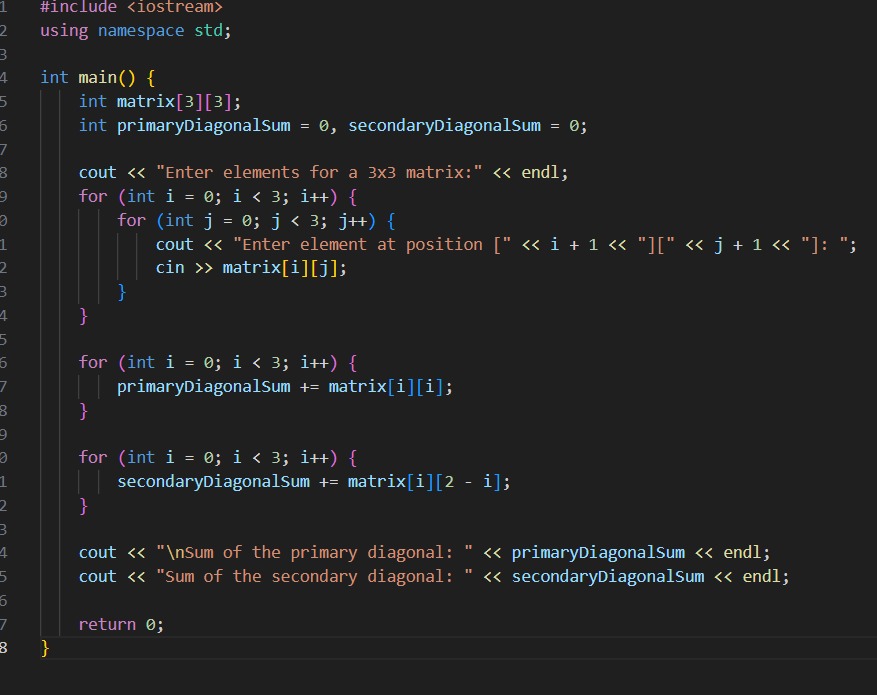
}

cout << "\nSum of the primary diagonal: " << primaryDiagonalSum << endl;

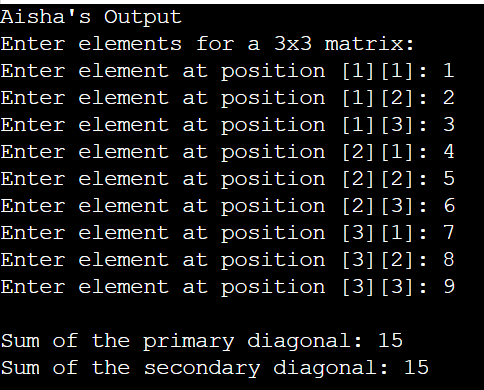
cout << "Sum of the secondary diagonal: " << secondaryDiagonalSum << endl;

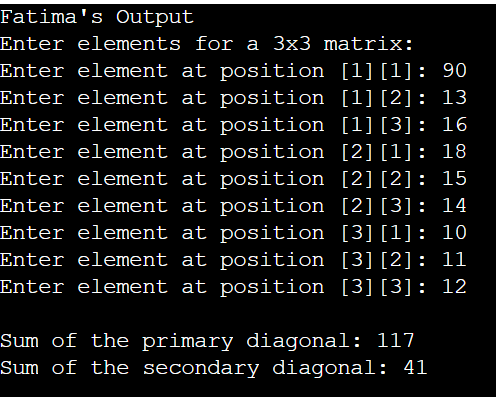
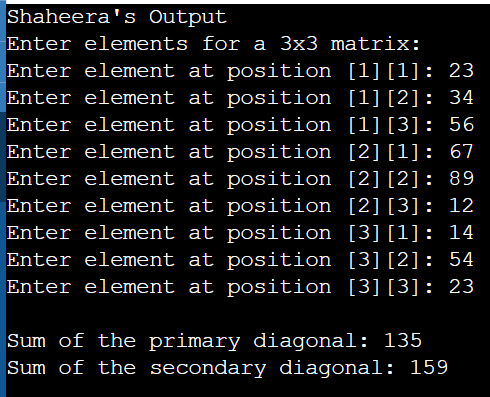
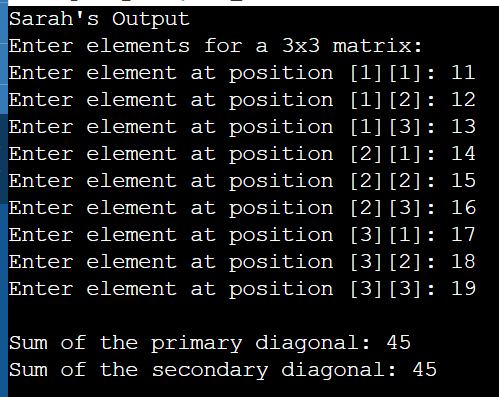
return 0;

}



**Output:**





**program2:**

**Write a program that implements both Linear Search and Binary Search.**

**• Create a sorted array of integers.**

**• Write two separate functions to implement Linear Search and Binary Search.**

**• The program should take an array and a target value as input and output whether the target value is found or not, and if found, its index.**

#include <iostream>

using namespace std;

int linearSearch(int arr[], int size, int target) {

for (int i = 0; i < size; i++) {

if (arr[i] == target) {

return i;

}

}

return -1;

}

int binarySearch(int arr[], int size, int target) {

int left = 0, right = size - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (arr[mid] == target) {

return mid;

}

else if (arr[mid] < target) {

left = mid + 1;

}

else {

right = mid - 1;

}

}

return -1;

}

int main() {

int size;

cout << "Enter the number of elements in the sorted array: ";

cin >> size;

int arr[size];

cout << "Enter " << size << " sorted integers:" << endl;

for (int i = 0; i < size; i++) {

cin >> arr[i];

}

int target;

cout << "Enter the target value to search: ";

cin >> target;

int linearResult = linearSearch(arr, size, target);

if (linearResult != -1) {

cout << "Linear Search: Target found at index " << linearResult << endl;

} else {

cout << "Linear Search: Target not found!" << endl;

}

int binaryResult = binarySearch(arr, size, target);

if (binaryResult != -1) {

cout << "Binary Search: Target found at index " << binaryResult << endl;

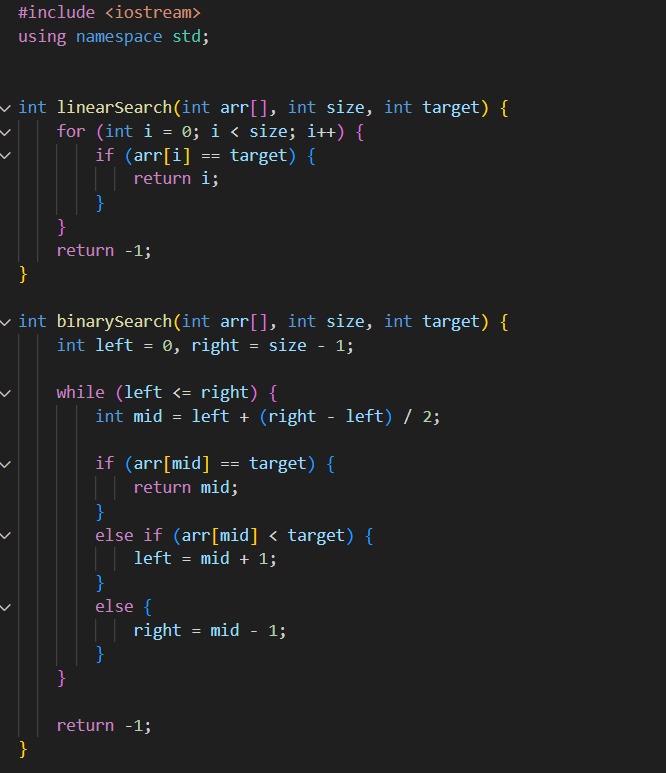
} else {

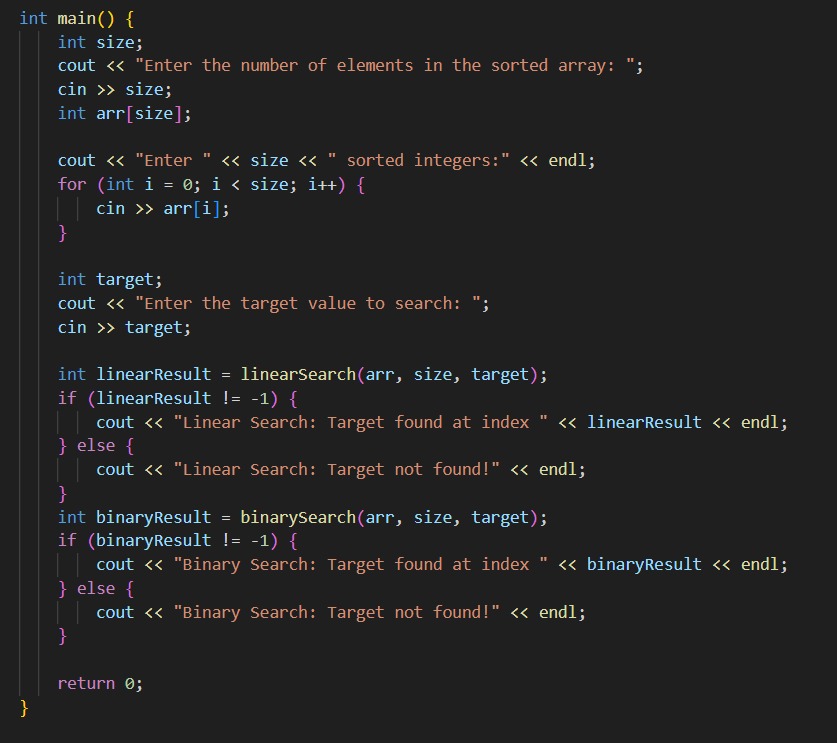
cout << "Binary Search: Target not found!" << endl;

}

return 0;

}





**Output:**

