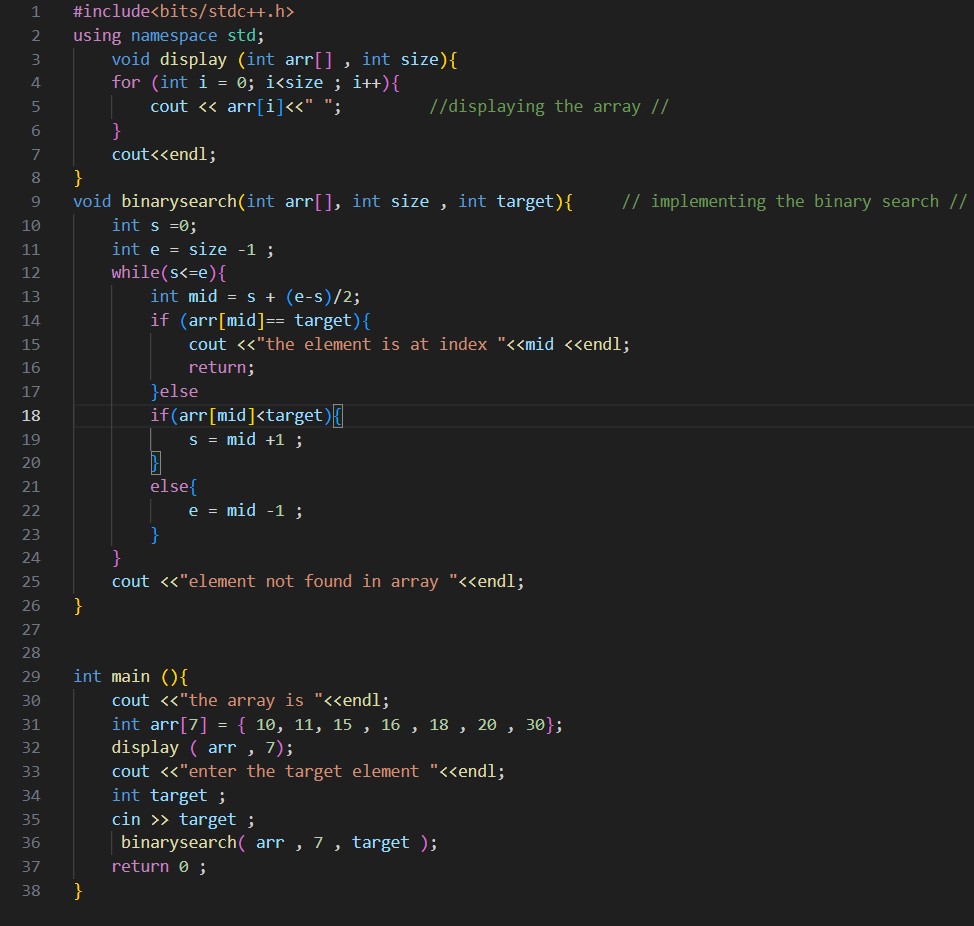
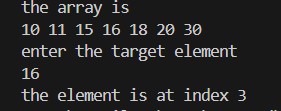
ASSIGNMENT 2:

NAME: Karanveer Singh ROLL NUMBER: 1024200002

QUE1: Implement the Binary search algorithm regarded as a fast search algorithm with run-time complexity of Ο(log n) in comparison to the Linear Search. CODE:



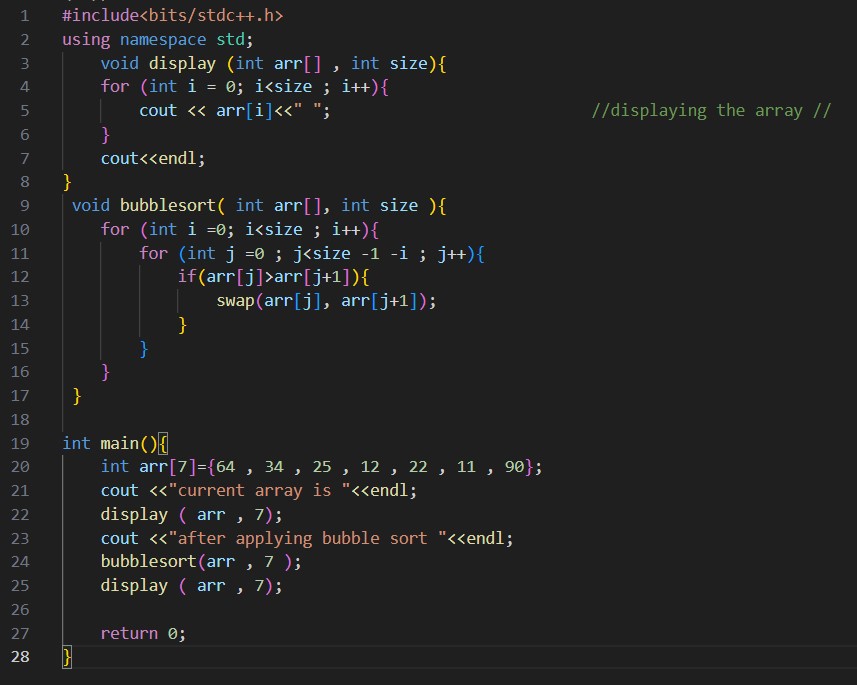


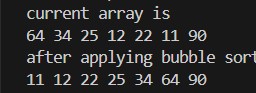
------------------X----------------------X----------------------X------------------------

QUE2: Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping

the adjacent elements if they are in wrong order. Code the Bubble sort with the following elements:

64 34 25 12 22 11 90 CODE:



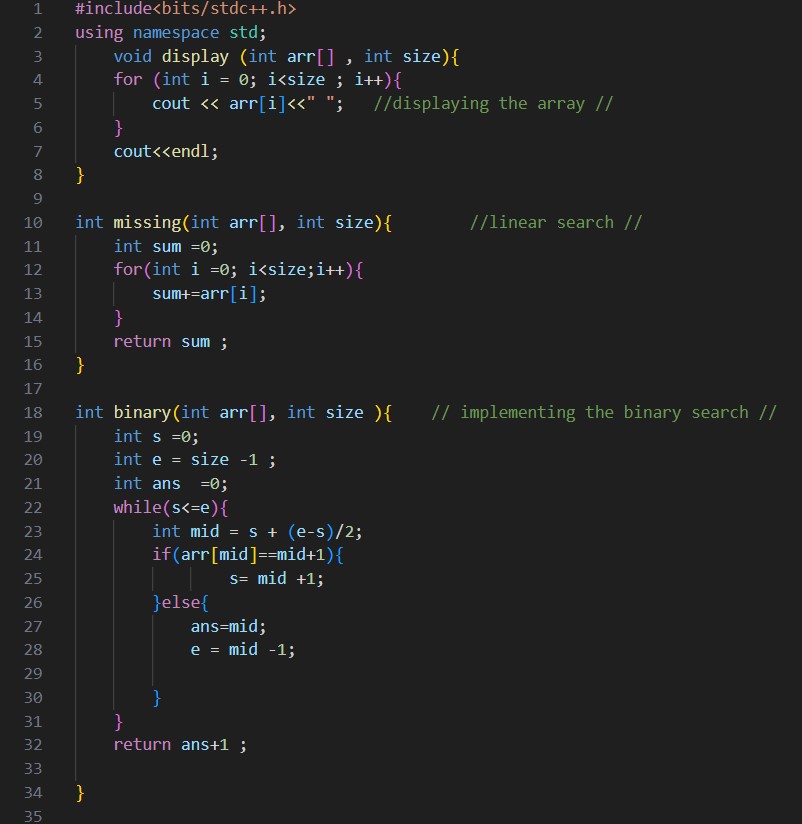


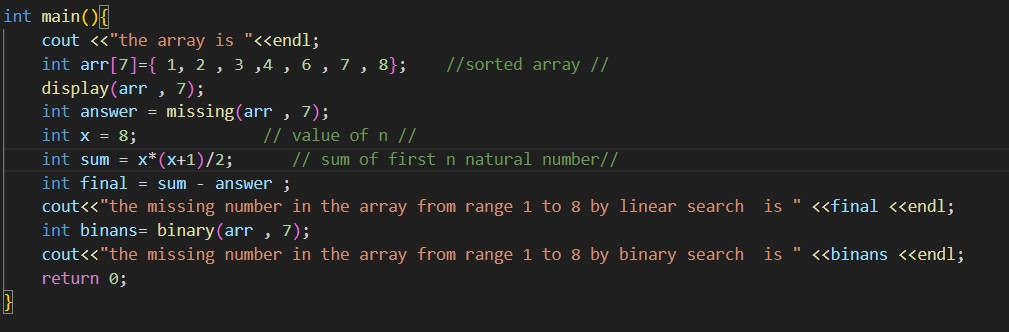
------------------X----------------------X----------------------X----------------------

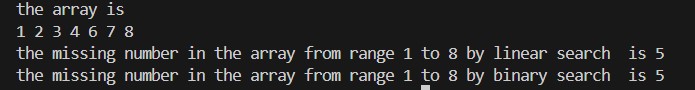
QUE3: Given an array of n-1 distinct integers in the range of 1 to n, find the missing number in it in a Sorted Array

1. Linear time
2. Using binary search.

CODE:





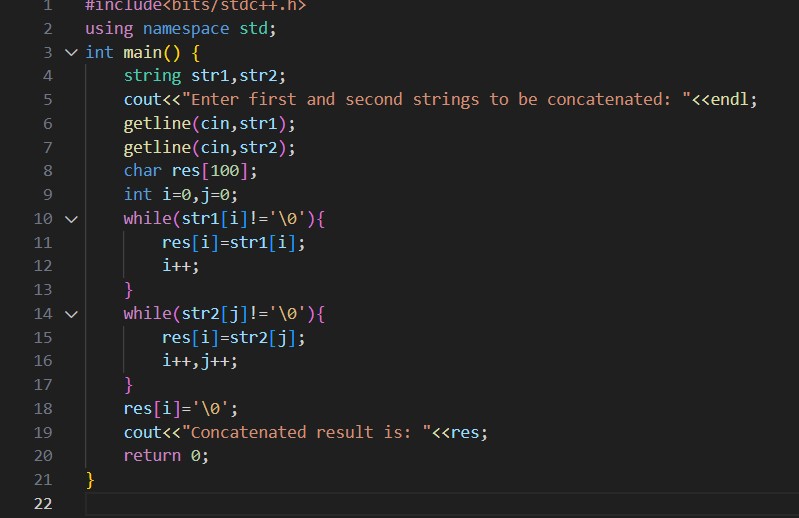


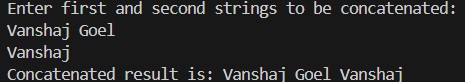
------------------X----------------------X----------------------X------------------------

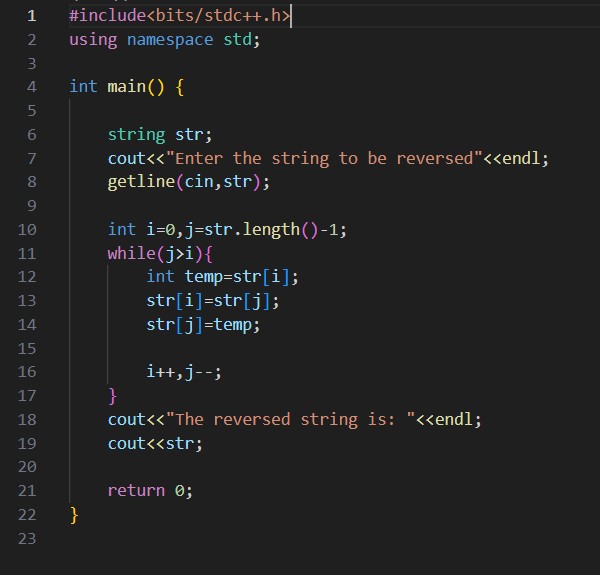
QUE4: String Related Programs

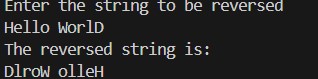
1. Write a program to concatenate one string to another string.
2. Write a program to reverse a string.
3. Write a program to delete all the vowels from the string.
4. Write a program to sort the strings in alphabetical order.
5. Write a program to convert a character from uppercase to lowercase.

CODE (a):

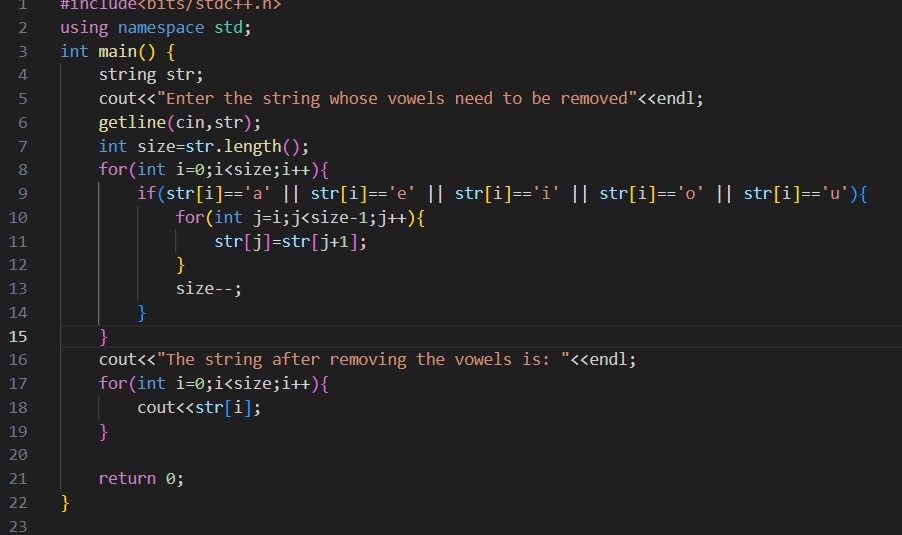


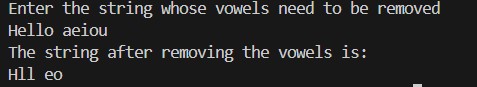
 CODE (b):



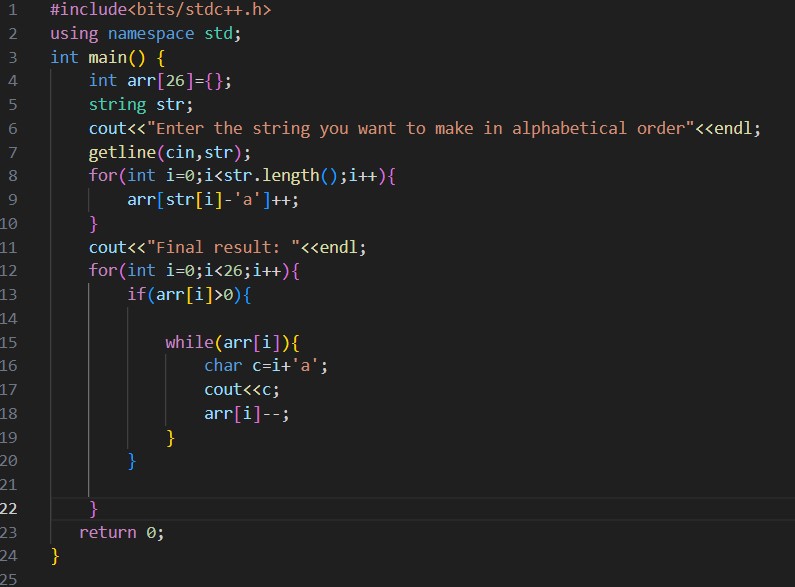


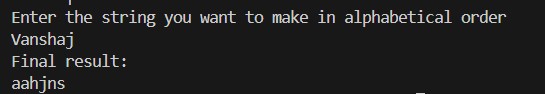
CODE (c):



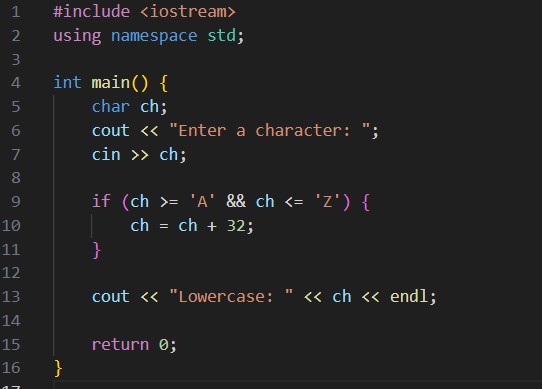


CODE (d):





CODE (e):



OUTPUT (e):

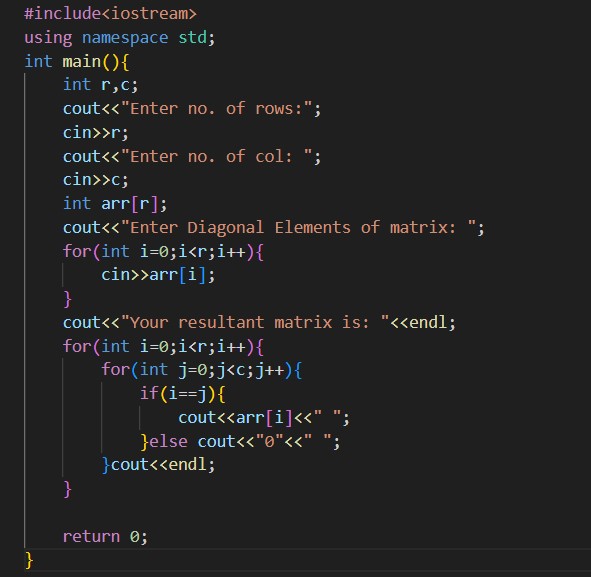


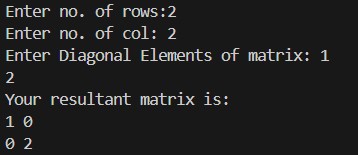
5) Space required to store any two-dimensional array is 𝑛𝑢𝑚𝑏𝑒𝑟 𝑜ƒ 𝑟𝑜𝑤𝑠 × 𝑛𝑢𝑚𝑏𝑒𝑟 𝑜ƒ c𝑜𝑙𝑢𝑚𝑛𝑠. Assuming an array is used to store elements of the following matrices, implement an efficient way that reduces the space requirement.

1. Diagonal Matrix.
2. Tri-diagonal Matrix.
3. Lower triangular Matrix. (d) Upper triangular Matrix.

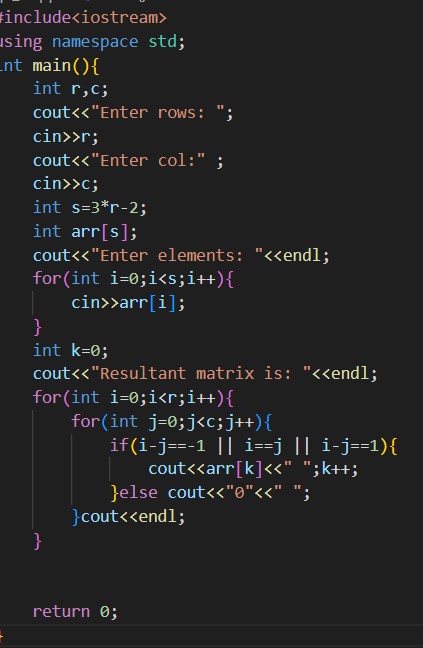
e) Symmetric Matrix

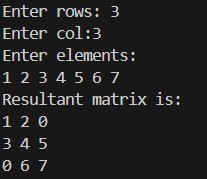
CODE A :



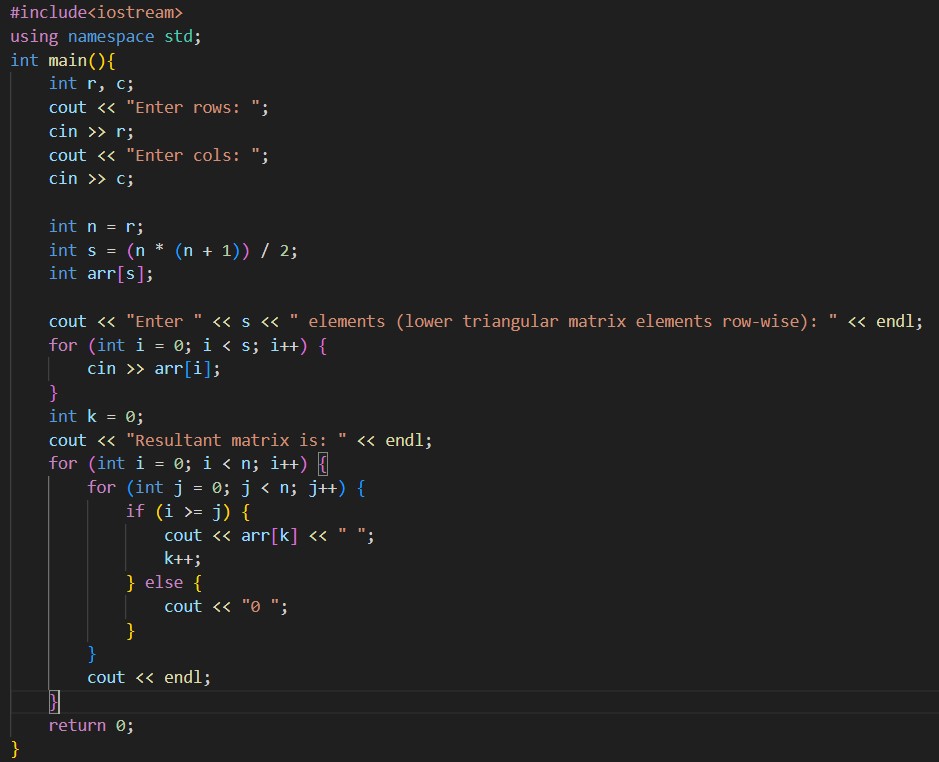


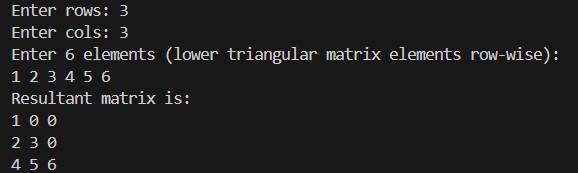
CODE B :



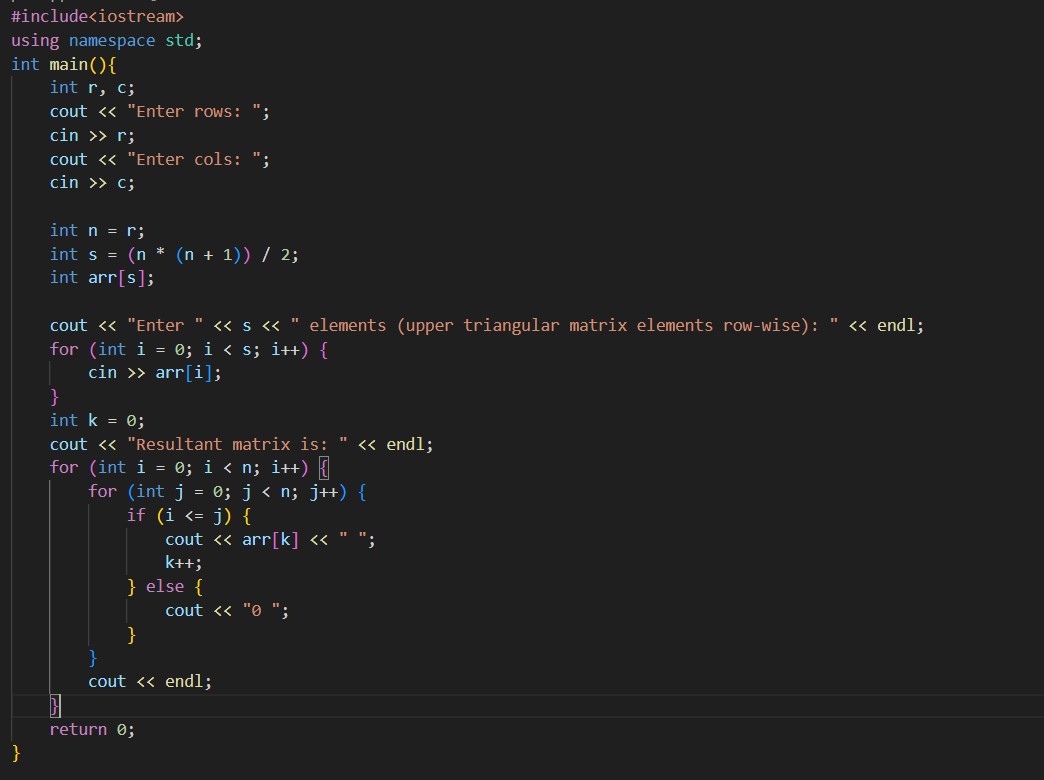


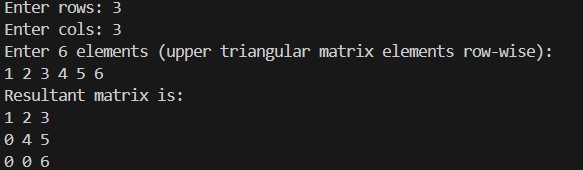
CODE C :



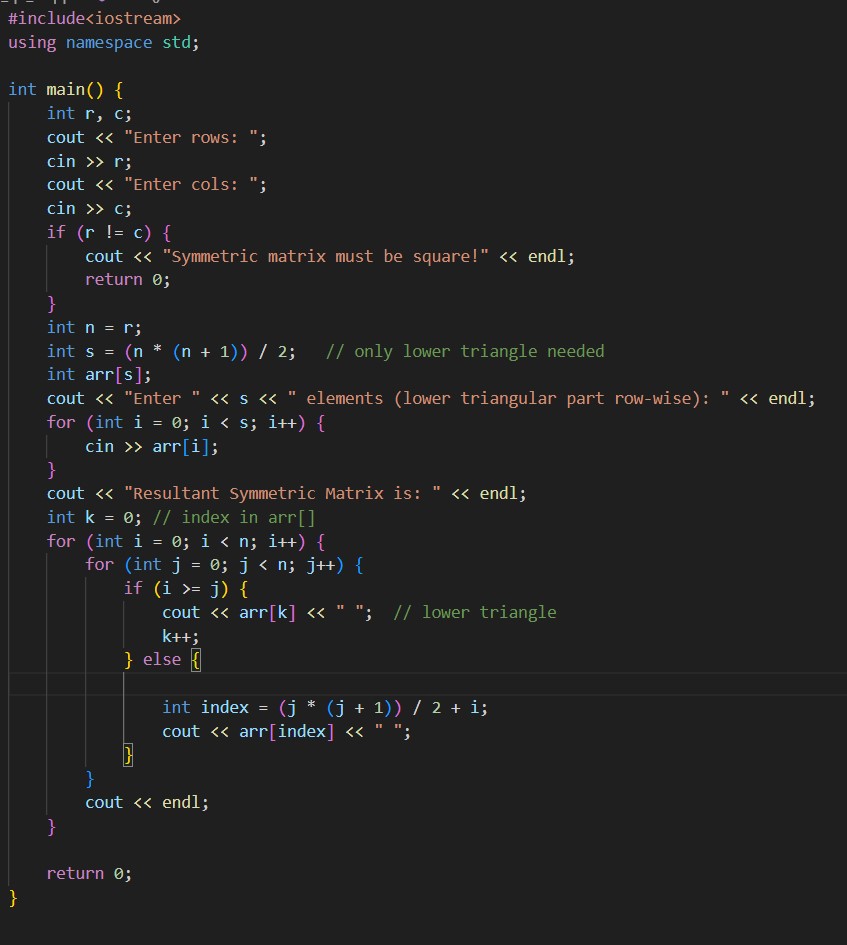


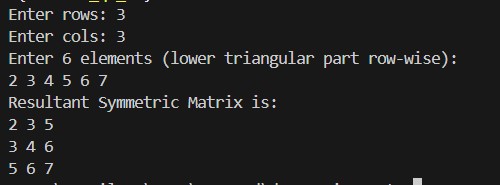
CODE 4 :





CODE 5 :

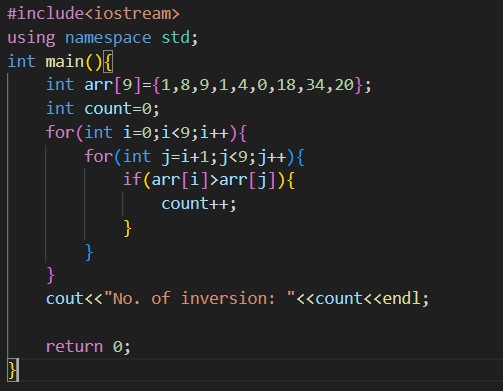




------------------X----------------------X----------------------X------------------------

QUE7: Let A[1 .... n] be an array of n real numbers. A pair (A[i], A[j ]) is said to be an inversion if these numbers are out of order, i.e., i < j but A[i]>A[j ]. Write a program to count the number of inversions in an array. CODE:

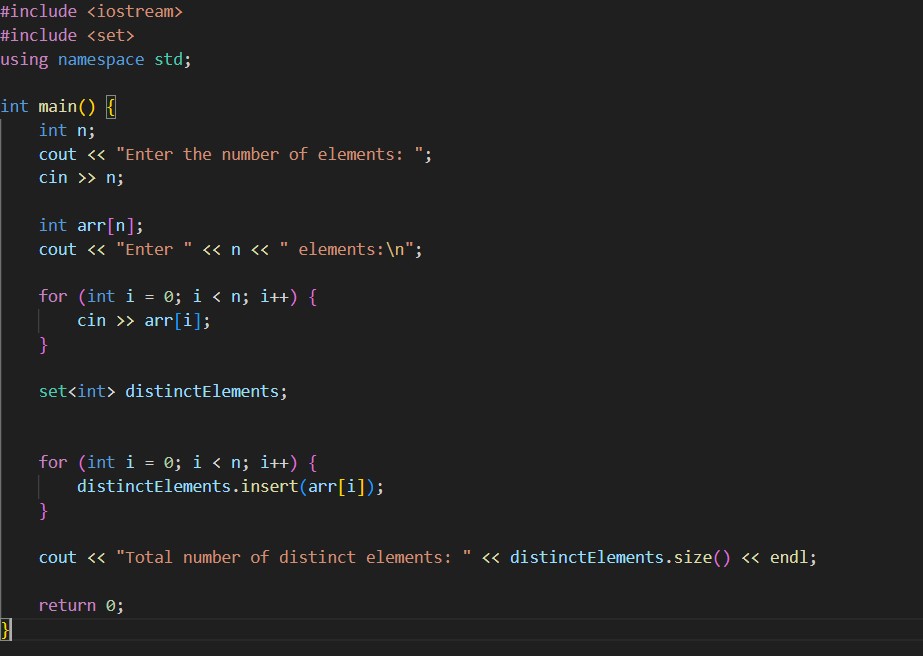
#include <iostream> using namespace std;

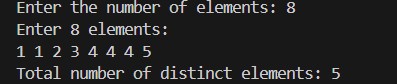


No. of Inversion in this case is 10.

------------------X----------------------X----------------------X------------------------

QUE8: Write a program to count the total number of distinct elements in an array of length n. CODE:





------------------X----------------------X----------------------X------------------------