

| **TITLE:**  Virtual Lab experiment on matrix multiplication |
| --- |

**AIM:** Virtual Lab experiment on recursion

<https://cse02-iiith.vlabs.ac.in/>

<https://cse02-iiith.vlabs.ac.in/exp/arrays/simulation.html>

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO3:** Illustrate the use of derived and structured data types such as arrays, strings, structures and unions.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Programming in C, second edition, Pradeep Dey and Manas Ghosh, Oxford University Press.
2. Programming in ANSI C, fifth edition, E Balagurusamy, Tata McGraw Hill.
3. Introduction to programming and problem solving , G. Michael Schneider ,Wiley India edition.
4. [**http://cse.iitkgp.ac.in/~rkumar/pds-vlab/**](http://cse.iitkgp.ac.in/~rkumar/pds-vlab/)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Problem Definition:**

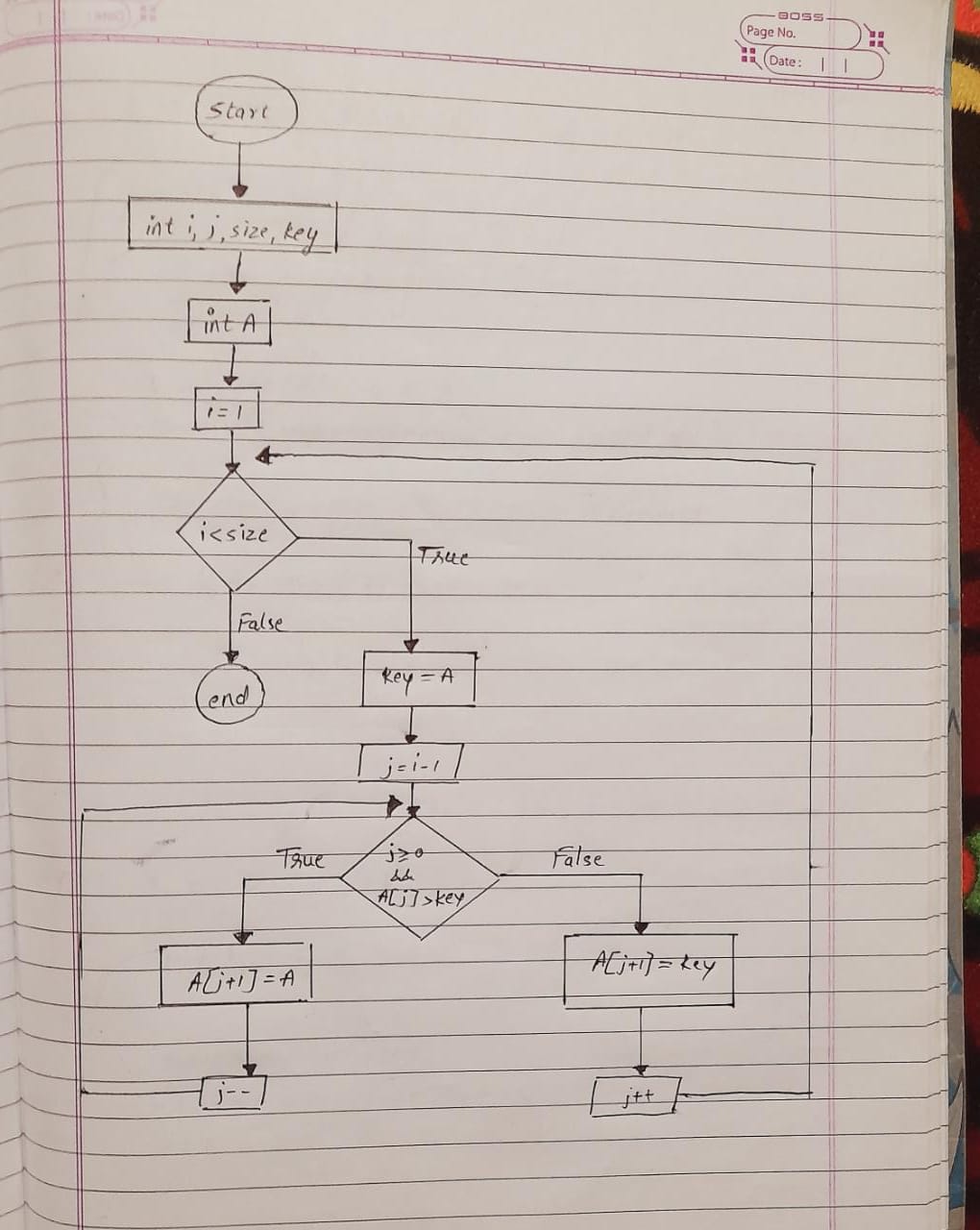
The Program implements 1-D and 2-D arrays.

1-D array prompts the user to enter the size of the array and elements. It then sorts the elements.

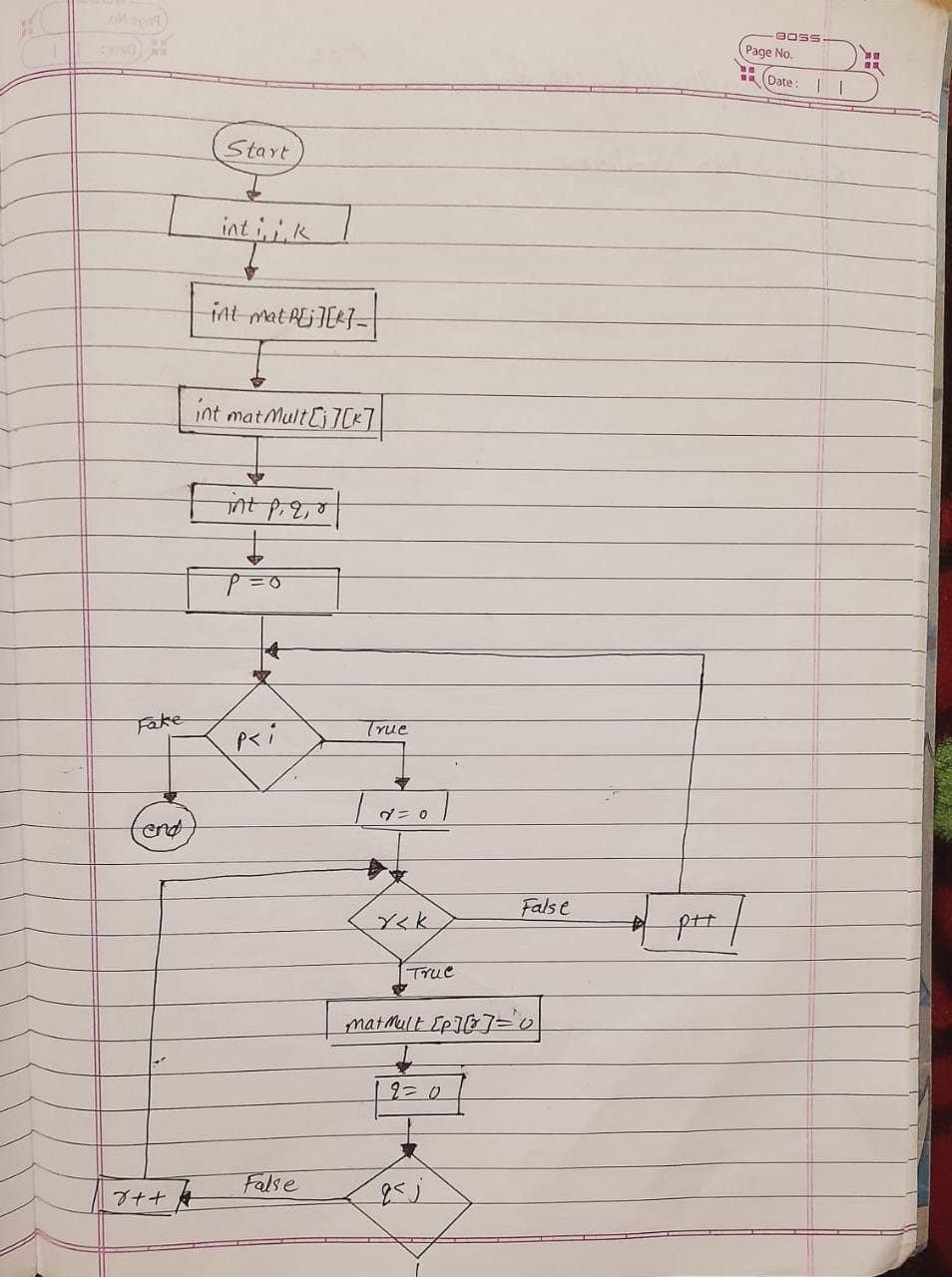
2-D array prompts the user to enter the order of two matrices and the elements. It then performs matrix multiplication.

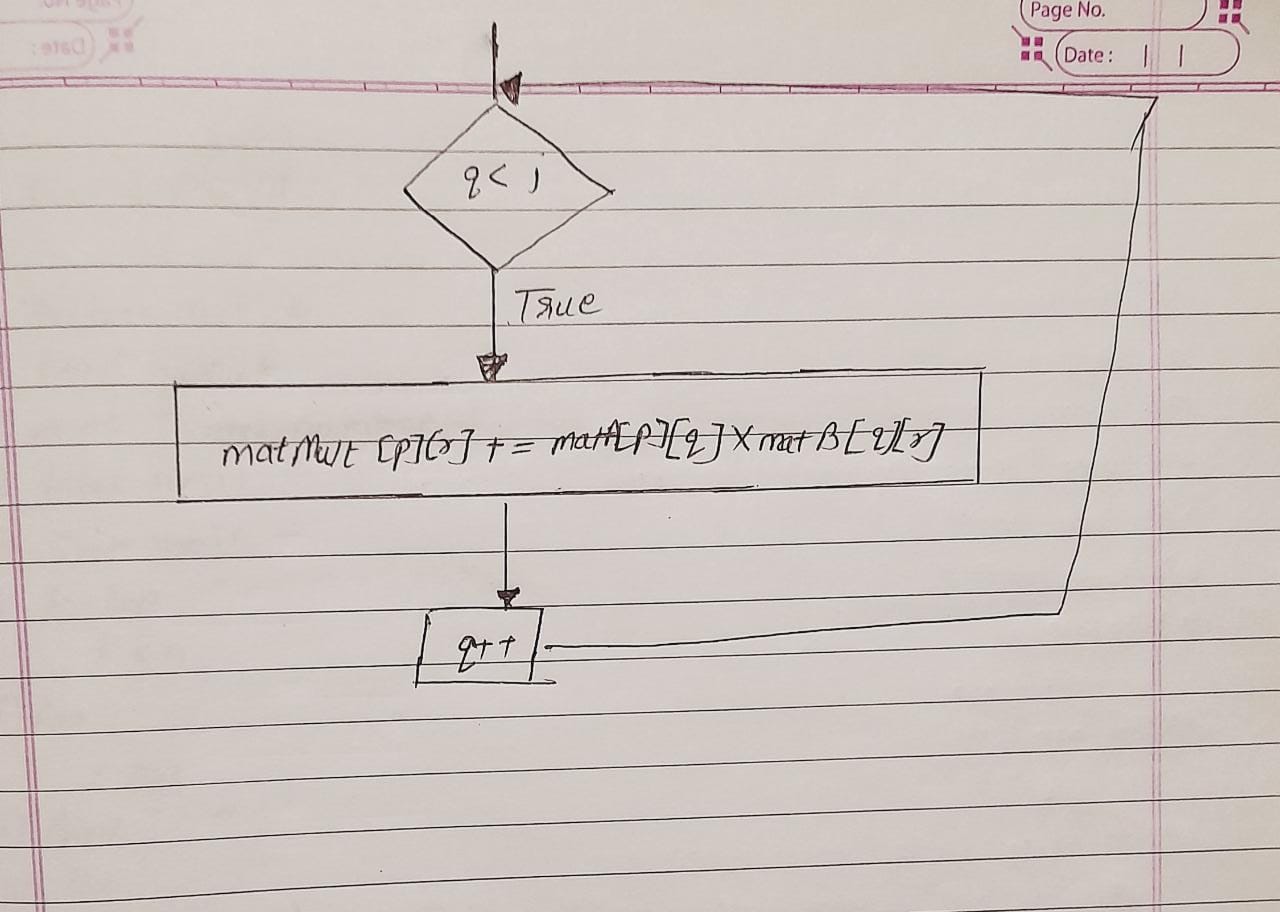
**Flowchart:**

**1D :-**

****

**2D:-**

****

****

**Implementation details:**

**1D:**

int main(){

int i, j, size, key ;

int A[size];

for( i = 1 ; i < size ; i++ )

{

key = A[i];

j = i - 1;

while ( j >= 0 && A[j] > key )

{

A[j+1] = A[j];

j--;

}

A[j+1] = key;

}

return 0 ;

}

**2D:**

int main(){

int i,j,k;

int matA[i][j];

int matB[j][k];

int matMult[i][k];

int p,q,r;

for ( p = 0 ; p < i ;p++ )

{

for ( r = 0 ; r <k ; r++ )

{

matMult[p][r] = 0;

for ( q = 0 ; q < j ; q++ )

{

matMult[p][r] += matA[p][q]\*matB[q][r]

}

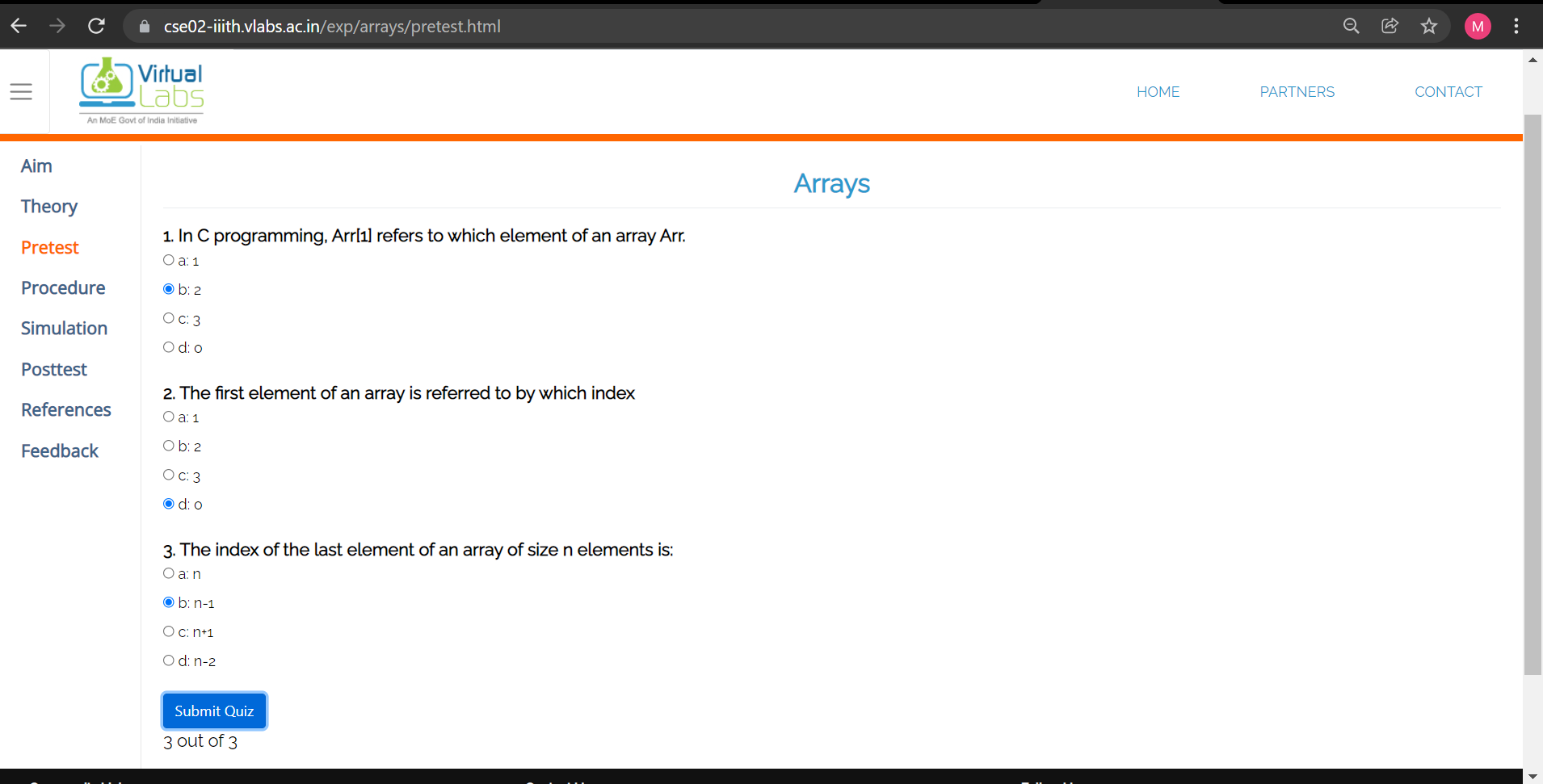
}

}

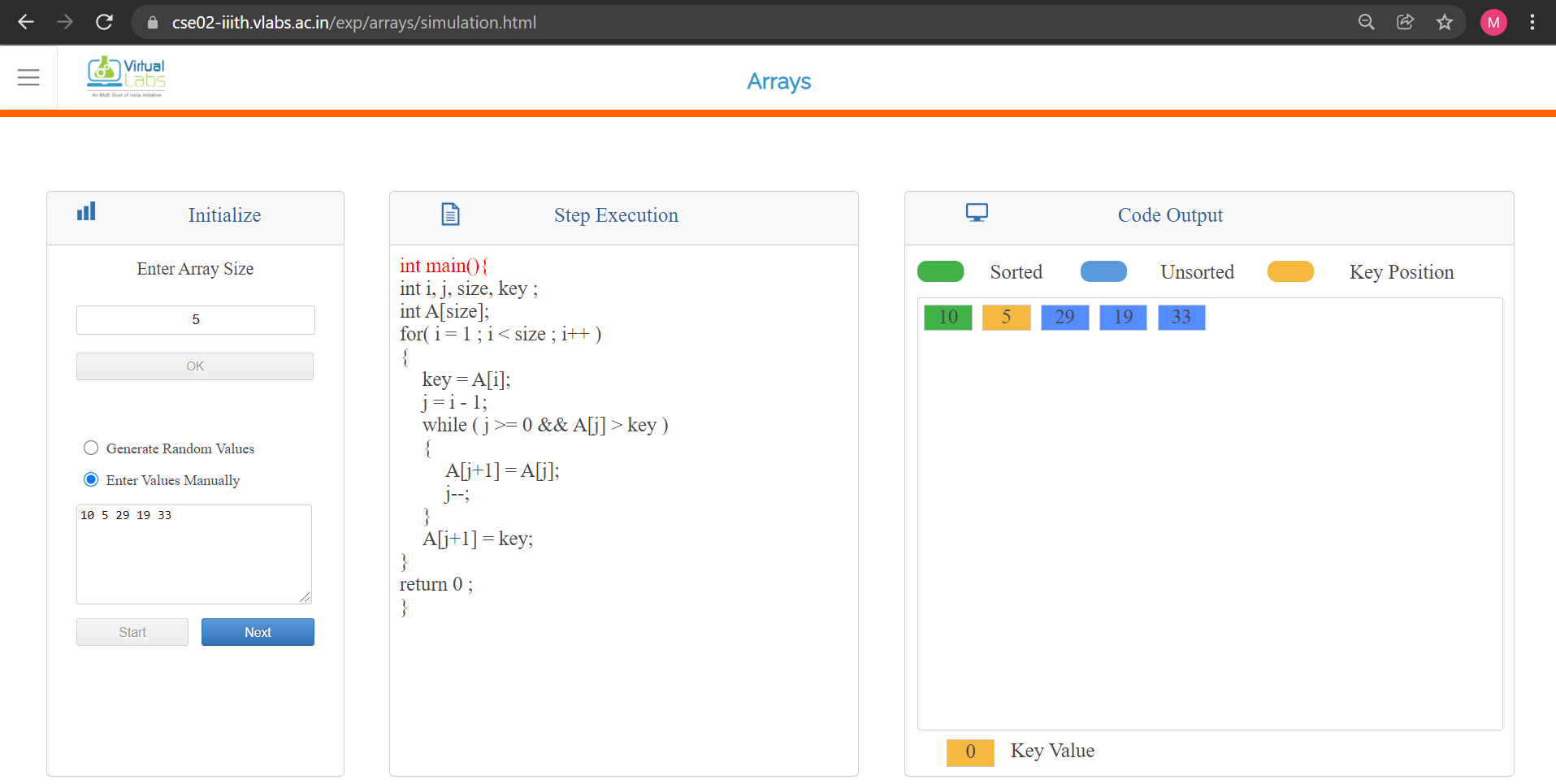
}

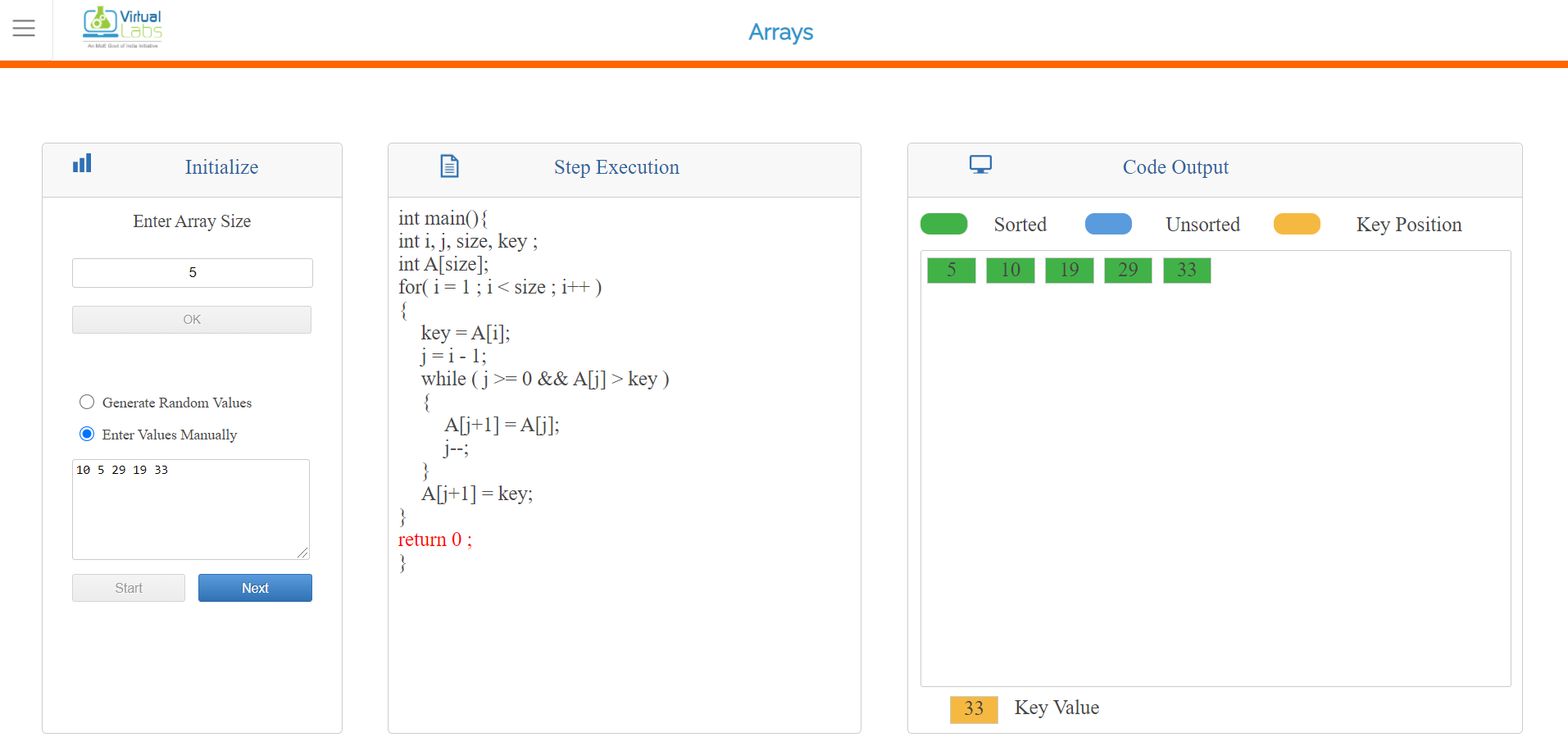
**Output(s):**

Pretest:

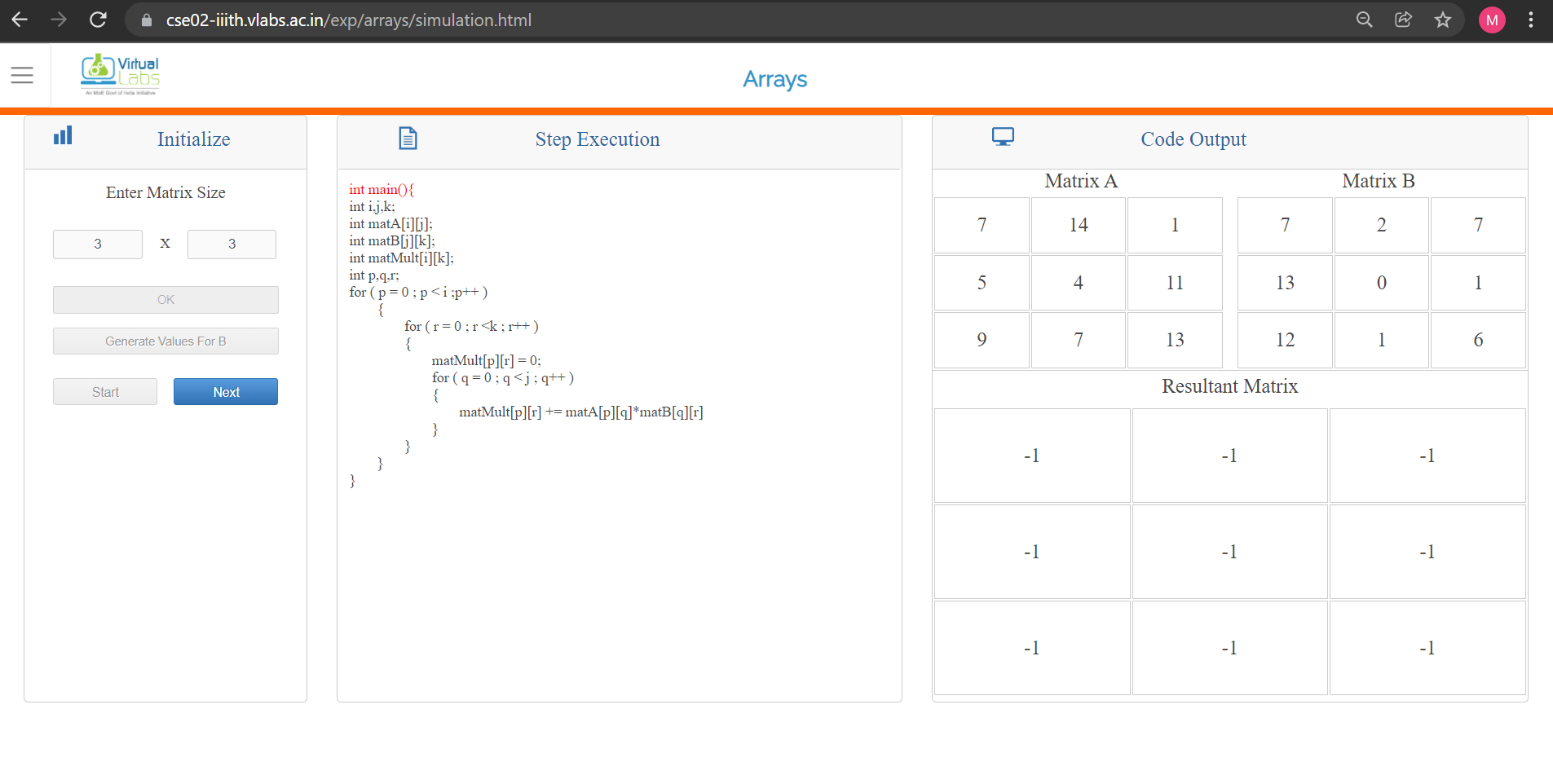


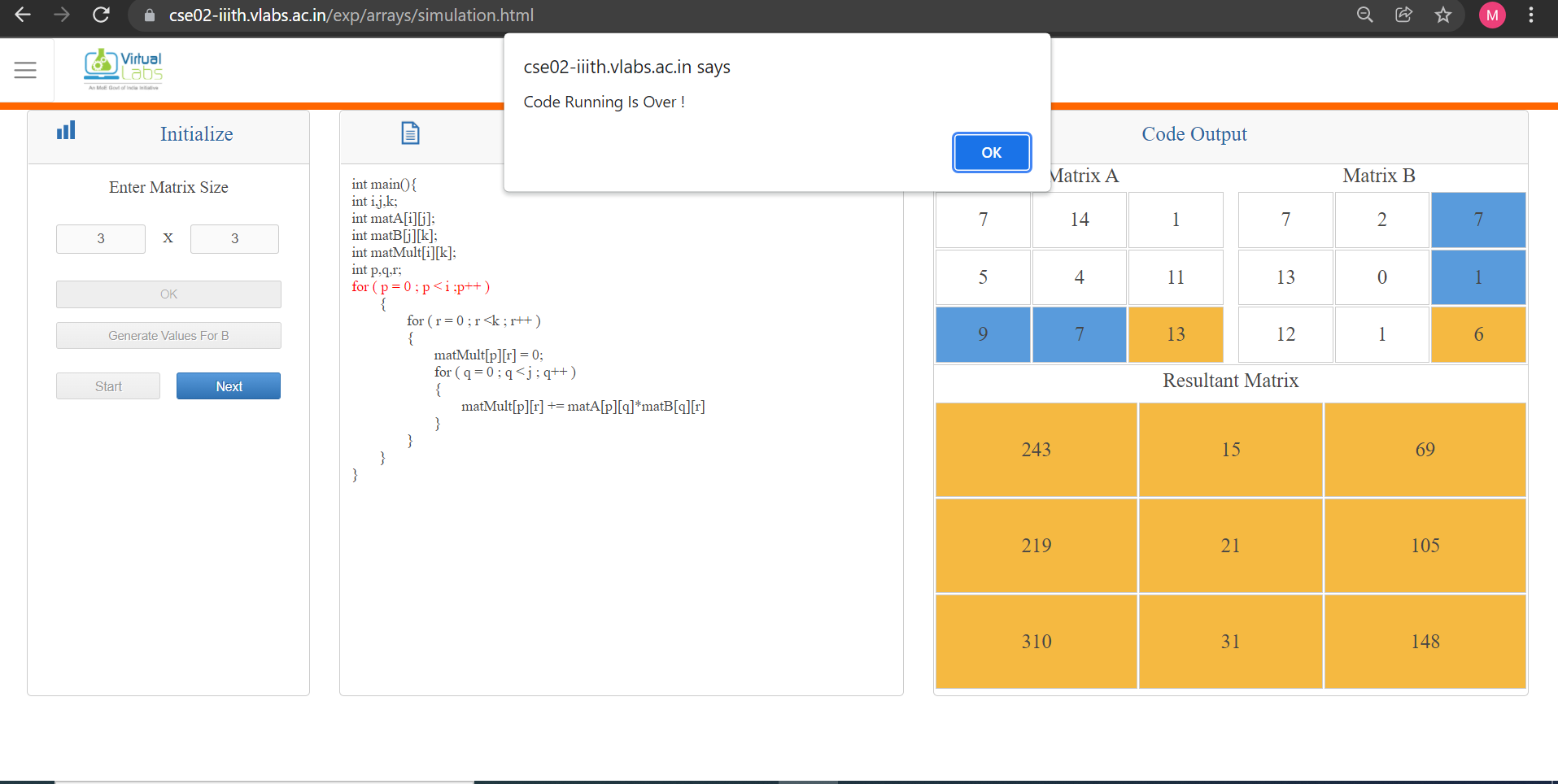
1D:



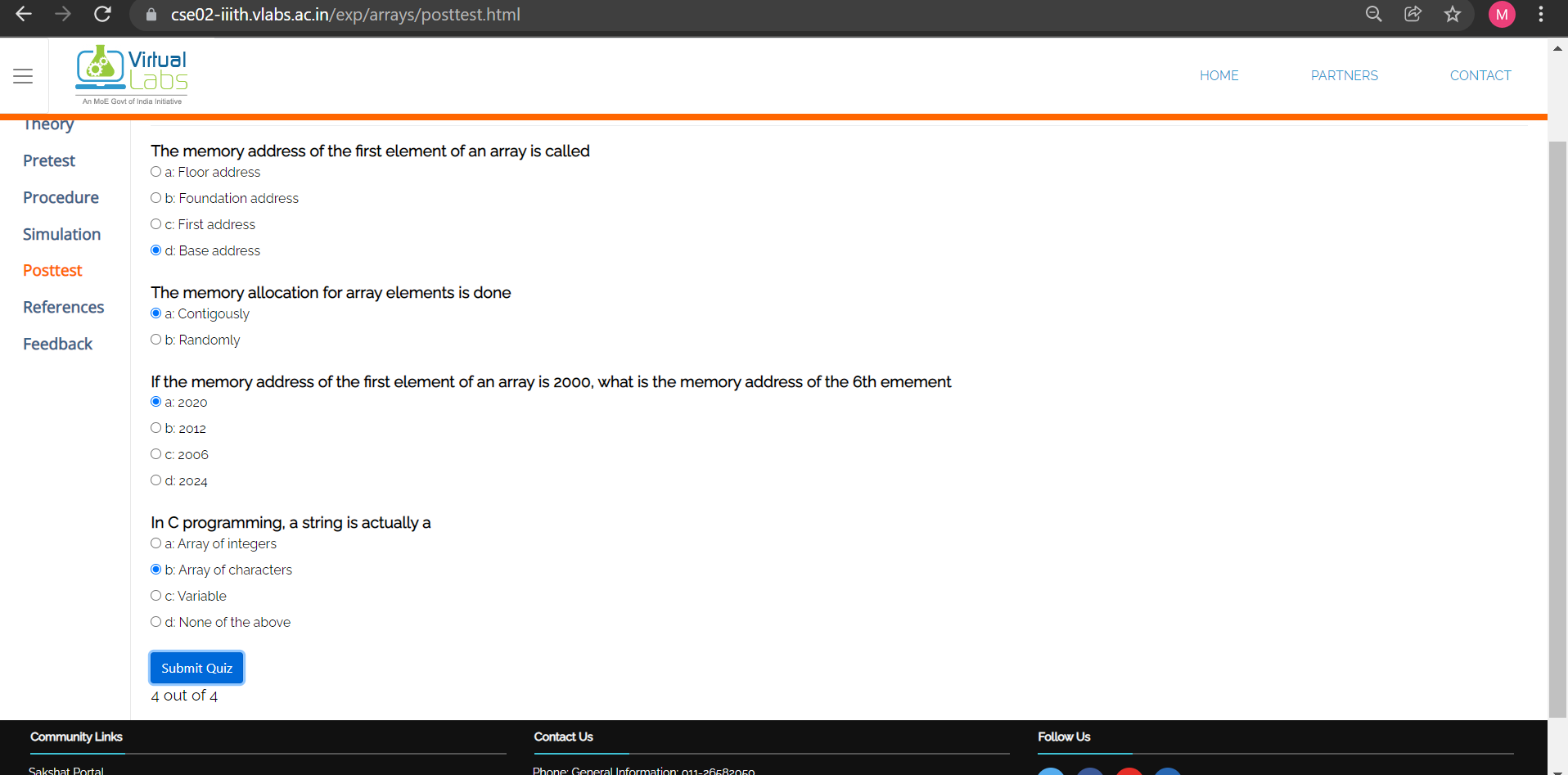


2D:





Posttest:



**Conclusion:**

The programs to sort a 1D array and to multiply two 2D arrays were learnt using online virtual simulation lab. The simulation worked perfectly.

**Post Lab Descriptive Questions**

1. **Can we change the size of an array at run time? Why or why not?**

**Ans :- For providing memory on a stack the size of the memory should be known to the compiler during compile time. So that during run time that much memory can be set aside for the variable on the stack. That is the reason you cannot decide the size of the array at run time as far as C language is concerned.**

1. **Can we pass an array as an argument to a function?**

**Ans :- There are two possible ways to do so, one by using call by value and other by using call by reference.**

**We can either have an array as a parameter. Example: int sum (int arr[]);**

**Or, we can have a pointers in the parameter list, to hold the base address of our array. Example: int sum (int\* ptr);**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**