



K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Batch: E1 Roll No.: 16010321011

Experiment / assignment / tutorial No. 7

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of the Staff In-charge with date

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:

For 1D array

Please enter the values of array

4,9,1,5,2

The sorted array is

1,2,4,5,9

For 2D array

Please enter values of first 2*2 matrix

7,12,13,2

Please enter the value of second 2*2 matrix

12,8,13,4

The matrix multiplication gives matrix

$$\begin{bmatrix} 240 & 104 \\ 182 & 112 \end{bmatrix}$$

Books/ Journals/ Websites referred:



K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

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/>

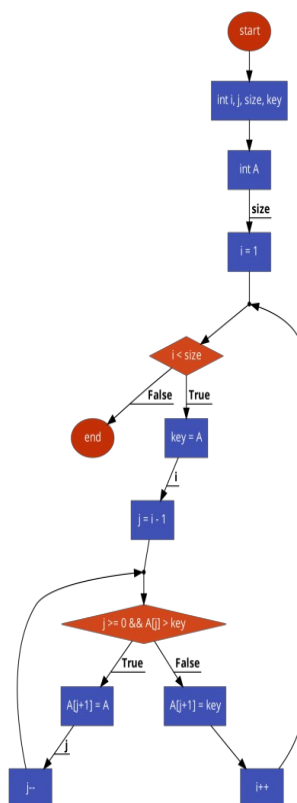
Problem Definition:

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: for 1D

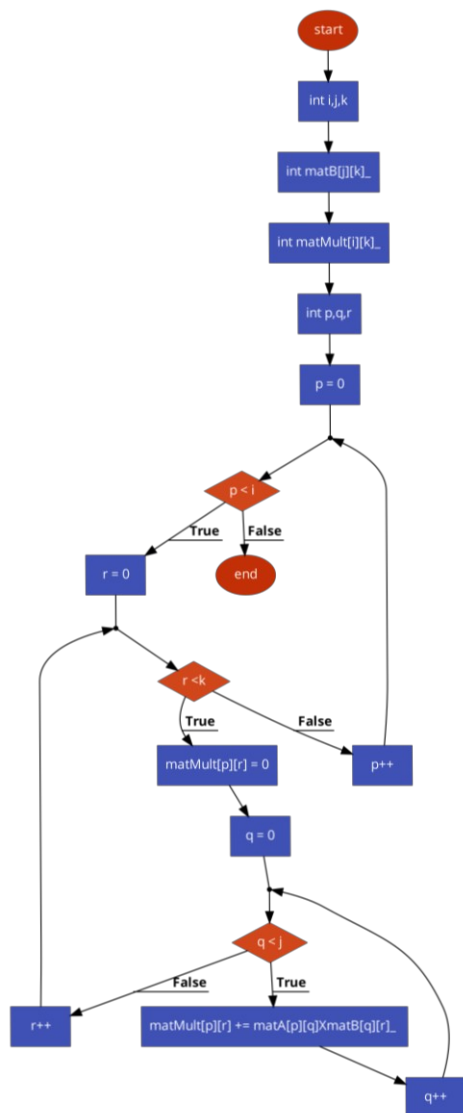


For 2D



K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)





K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Implementation details:

For 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 ;
}
```

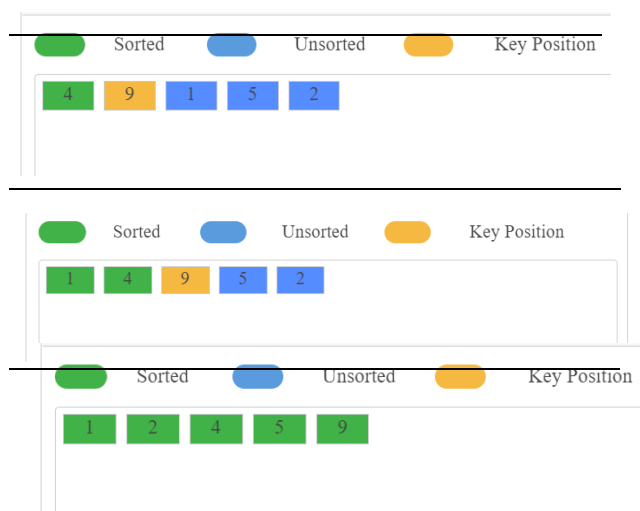
For 2D

```
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):

(Attach screenshots of the Output of Program Code implemented in Virtual Lab and Quiz attempted)

For 1D



For 2D



K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Code Output			
Matrix A		Matrix B	
7	12	12	8
13	2	13	4
Resultant Matrix			
-1		-1	
-1		-1	

Code Output			
Matrix A		Matrix B	
7	12	12	8
13	2	13	4
Resultant Matrix			
240		104	
182		112	



K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)

Conclusion:

Pre-test:

1. In C programming, Arr[1] refers to which element of an array Arr.

☐ a: 1

☒ b: 2

☐ c: 3

☐ d: 0

2. The first element of an array is referred to by which index

☐ a: 1

☐ b: 2

☐ c: 3

☒ d: 0

3. The index of the last element of an array of size n elements is:

☐ a: n

☒ b: n-1

☐ c: n+1

☐ d: n-2

[Submit Quiz](#)

3 out of 3

Post-test:

The memory address of the first element of an array is called

☐ a: Floor address

☐ b: Foundation address

☐ c: First address

☒ d: Base address

The memory allocation for array elements is done

☒ a: Contiguously

☐ b: Randomly

If the memory address of the first element of an array is 2000, what is the memory address of the 6th element

☒ a: 2020

☐ b: 2012

☐ c: 2006

☐ d: 2024

In C programming, a string is actually a

☒ a: Array of integers

☐ b: Array of characters

☐ c: Variable

☐ d: None of the above

[Submit Quiz](#)

4 out of 4



K. J. Somaiya College of Engineering, Mumbai-77

(A Constituent College of Somaiya Vidyavihar University)

Post Lab Descriptive Questions

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

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.

Hence we cannot decide the size of the array at run time as far as C language is concerned.

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

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

*array as a parameter: Example: `int sum (int arr[]);`

*pointers in the parameter list, to hold the base address of our array

:Example: `int sum (int* ptr);`

Date: _____

Signature of faculty in-charge