Fundamentals of Programming Using C - 15101104

Unit 3 : Array

Bhavika Vaghela

Assistant Professor

Parul Institute of Computer Application

Faculty of IT & Computer Science

Parul University

Brief about Array

What is array?

- Array is derived data structure in C programming which is use to store data of same type in a continuous manner.
- Array store data of same data type (homogenous) mean if one want to store marks of 50 students or more than it is possible using array.
- Index value of array start with “0”.
- C support two types of array

1) One dimensional
2) Multi dimensional array.

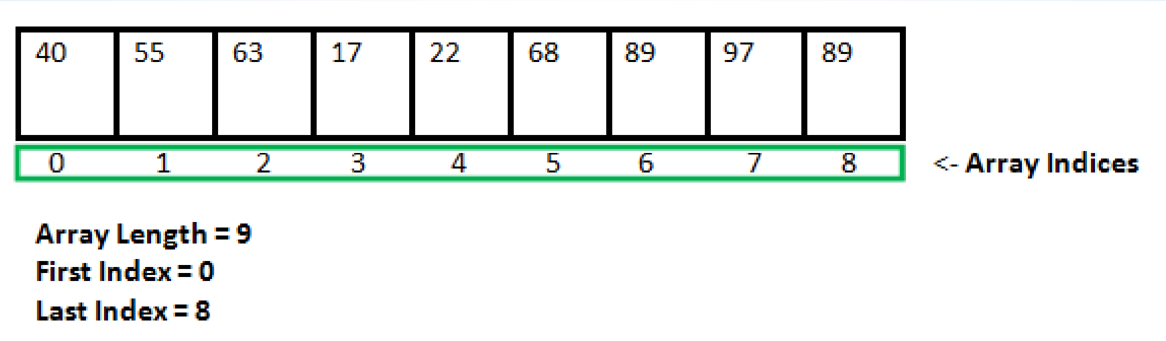
- Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc.

Cont…

- As per last slide if one wants to store marks of 50 students than rather to declare 50 variable it's better to declare one variable and store 50 value.
- So that one can access all 50 value using one variable easily by iterating it using loop.
- An array is a fixed size sequenced collection of element of the same data type.
- In simple way, an array can be used to represent a list of numbers, or list of names.
- Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

Cont…

- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



40	55	63	17	22	68	89	97	89	
0	1	2	3	4	5	6	7	8	<- Array Indices

Array Length = 9
First Index = 0
Last Index = 8

Properties of array

1) Element of array is of same data type.
2) It store in continuous memory.
3) We can access it in sequence or randomly by passing it index value.

Cont…

Advantages

- **Code Optimization**: Less code to the access the data.
- **Ease of traversing**: By using the for loop, we can retrieve the elements of an array easily.
- **Ease of sorting**: To sort the elements of the array, we need a few lines of code only.
- **Random Access**: We can access any element randomly using the array.

Disadvantage

- **Fixed Size:** Whatever size, we define at the time of declaration of the array, we can't exceed the limit. So, it doesn't grow the size dynamically like LinkedList which you will learn later.

Cont…

- As discuss earlier C support two types of array
  - One dimensional
  - Multi dimensional array
    - Two dimension
    - Three dimension
    - Four dimension
- But we only learn one dimensional and two dimensional array.

How to declare array?

Syntax : <datatype> <array name> [Size];

```
int marks[10];
```

Cont…

How to initialize array?

1) At the time of declaration like ,
int marks[] = {10,12,24,25,9}; // size 5 automatically assign **int marks[5]={20,30,40,50,60};**

2) **After declaration like,**

int marks[5];

marks[0] = 12

marks[1] = 13

marks[2] = 10

marks[3] = 11

marks[4] = 20

Program to store and print element in 1D array

How to store name using array or word using array?

- To store name of word using array we have to declare array of char data type.
- Because if we declare variable like **char ch;** than it only allow to store single character like ‘a’, ‘A’, ‘&’, ‘+’ or any other character.
- But to store name have declare array of character datatype like
- char name[10];   //it allow to store 10 character long name.
- If one wants to store “Good Morning” then declare array of length 11 because white space is also known as a character constant.

e.g. **char name[]=“Kelvin”**   //length automatically assign by 6

One can also assign using pointer. About pointer will be discuss later on.

Program using 1D array

1) Program to store 10 number and print it.
2) Program to store 10 number and do addition of it.
3) Program to store n number and find maximum number from it.
4) Program to store n number and sort it into ascending order.
5) Program to store string in 1D array and print it.

Implement below by your own.

6) Program to store n numbers and perform multiplication of it.
7) Program to store n numbers and sort it into descending order.
8) Program to store n numbers and find minimum number from it.
9) Program to store n numbers and find number is present in array or not also print it's index value if number is present.

2 D array (two dimensional array)

- 2D array is combination of raw and column.
- Like 1D array index value of 2D array also begin with “0” indices (index).
- It is represented in the form of rows and columns, also known as matrix.
- It is also known as array of arrays or list of arrays.
- The two dimensional, three dimensional or other dimensional arrays are also known as multidimensional arrays.
- Declaration of two dimensional Array in C.

data_type array_name[size1][size2];

int twodimen[4][3]; 4 rows and 3 column

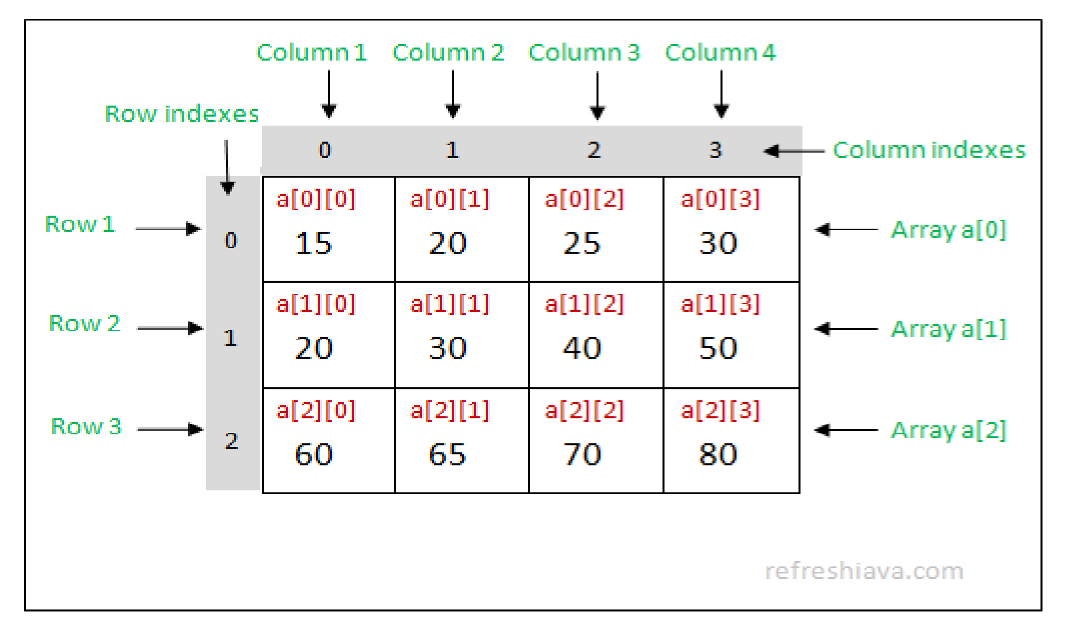
Declaration and initialization of 2D array

- int twodimen[4][3]; 4 rows and 3 column
- int num2d [ ] [ ] = {{1,2,3}, {4,5,6},{7,8,9}} its 3 by 3 array and it have total 9 elements.
- One can also declare at run time by iterating loop.

Program to declare and initialize and print 2D array at runtime.

Index representation

of 2D array



Column 1 Column 2 Column 3 Column 4

Row indexes

Column indexes

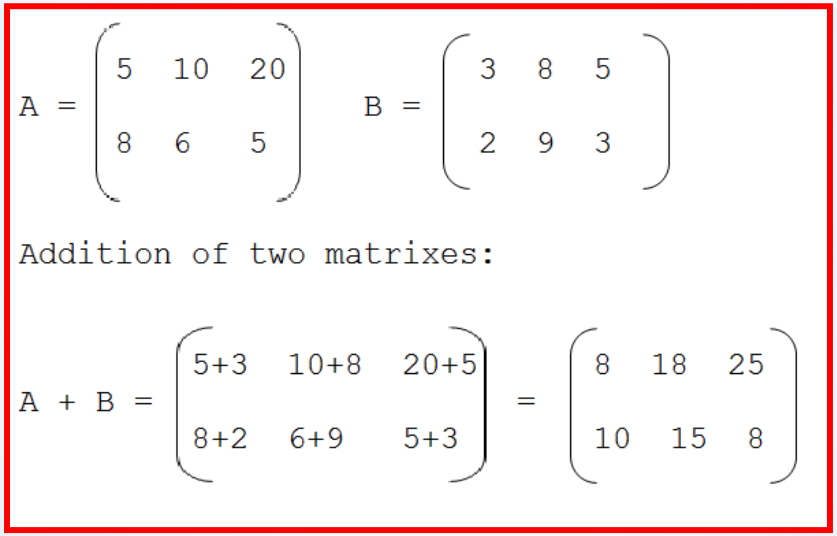
	0	1	2	3	
Row 1 → 0	a[0][0] 15	a[0][1] 20	a[0][2] 25	a[0][3] 30	Array a[0]
Row 2 → 1	a[1][0] 20	a[1][1] 30	a[1][2] 40	a[1][3] 50	Array a[1]
Row 3 → 2	a[2][0] 60	a[2][1] 65	a[2][2] 70	a[2][3] 80	Array a[2]

refreshjava.com

Addition of two 2D matrix :

- Rule: Addition of two matrices is only possible if both matrices are of same size.
- Suppose two matrices A and B is of same size m X n
- Sum of two matrices is defined as

$$(A + B)ij = Aij + Bij$$



$$A = \begin{pmatrix} 5 & 10 & 20 \\ 8 & 6 & 5 \end{pmatrix} \qquad B = \begin{pmatrix} 3 & 8 & 5 \\ 2 & 9 & 3 \end{pmatrix}$$

Addition of two matrixes:

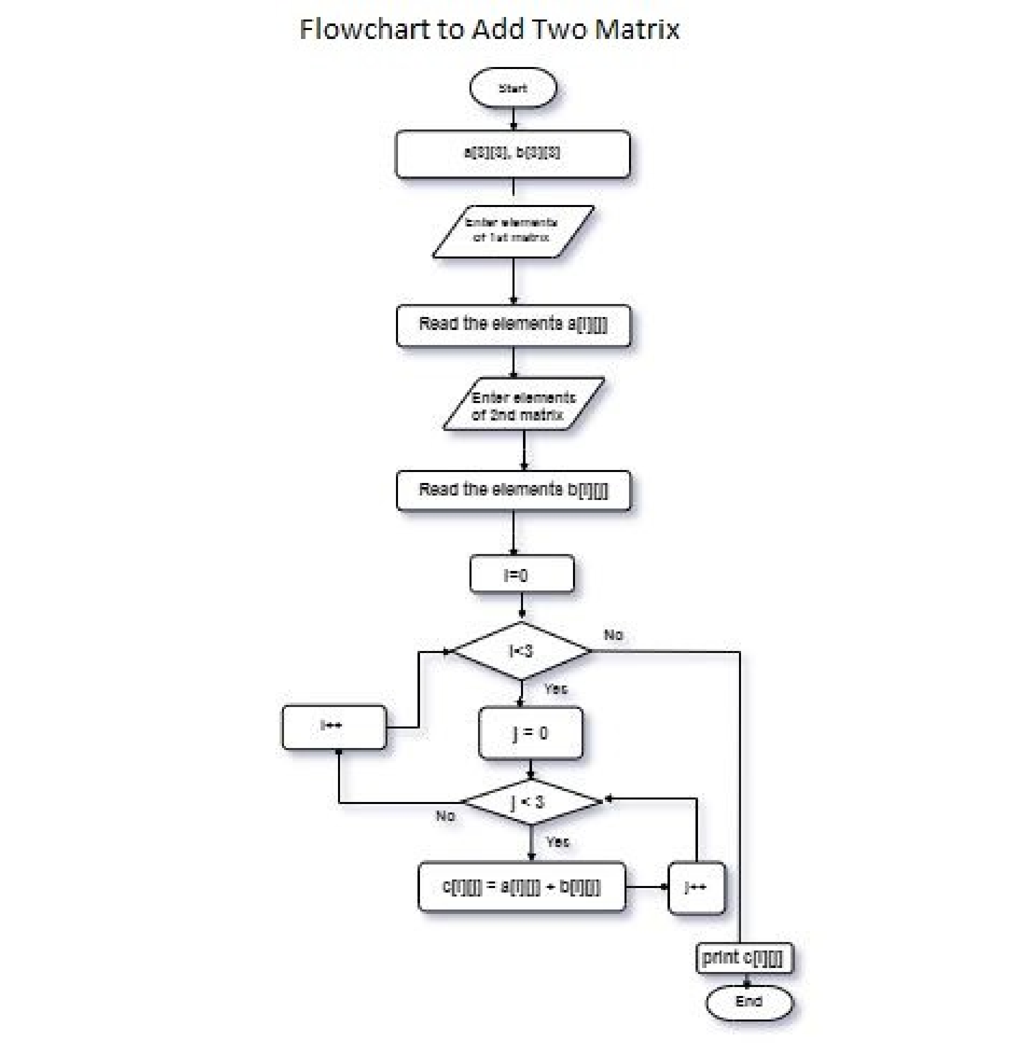
$$A + B = \begin{pmatrix} 5+3 & 10+8 & 20+5 \\ 8+2 & 6+9 & 5+3 \end{pmatrix} = \begin{pmatrix} 8 & 18 & 25 \\ 10 & 15 & 8 \end{pmatrix}$$

Algorithm for addition of two matrix

```
Step1: Start
Step2: Read: m and n
Step3: Read: Take inputs for Matrix A[1:m, 1:n] and Matrix B[1:m, 1:n]
Step4: Repeat for i := 1 to m by 1:
                Repeat for j := 1 to n by 1:
                        C[i, j] := A[i, j] + B[i, j]
                [End of inner for loop]
        [End of outer for loop]
Step5: Print: Matrix C
Step6: Exit.
```

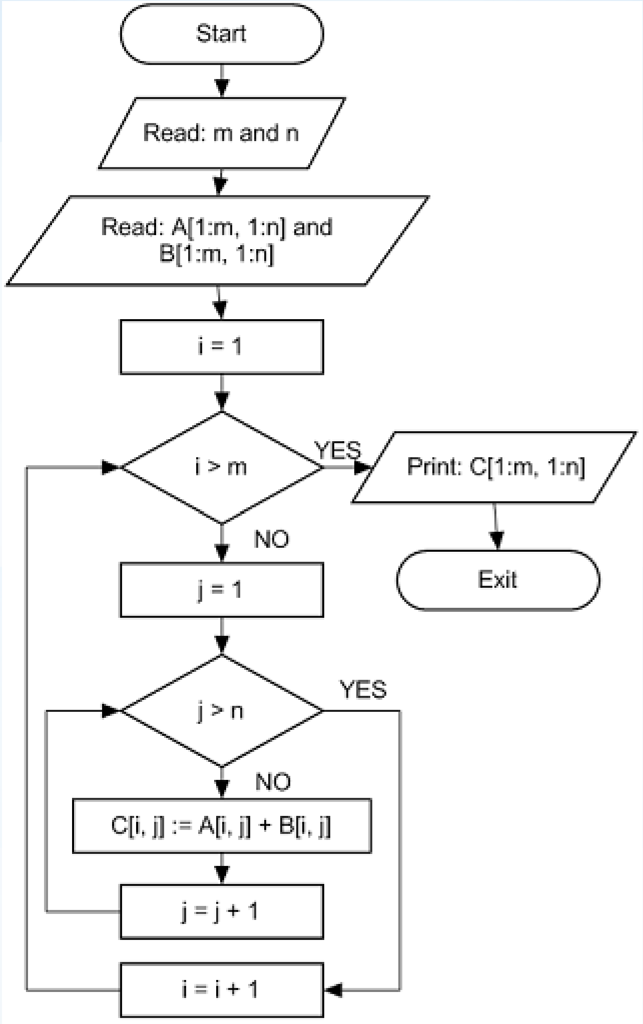
Program for matrix addition

Flowchart for addition of two matrix


Flowchart to Add Two Matrix

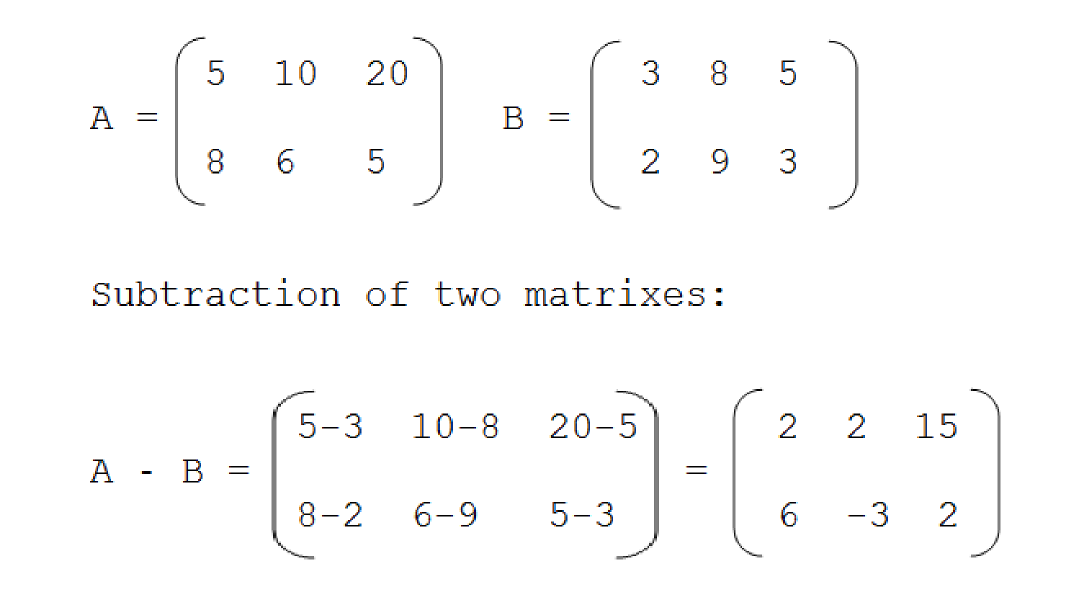
Start
a[3][3], b[3][3]
Enter elements of 1st matrix
Read the elements a[i][j]
Enter elements of 2nd matrix
Read the elements b[i][j]
i=0
i<3
No
Yes
i++
j = 0
j < 3
No
Yes
c[i][j] = a[i][j] + b[i][j]
j++
print c[i][j]
End

Flowchart for addition of two matrix



Subtraction of Matrix (2D array)

- Rule: Subtraction of two matrixes is only possible if both matrixes are of same size.
- Suppose two matrixes A and B is of same size m X n Subtraction of two marixes is defined as
- (A - B)ij = Aij - Bij



$$A = \begin{pmatrix} 5 & 10 & 20 \\ 8 & 6 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 8 & 5 \\ 2 & 9 & 3 \end{pmatrix}$$

Subtraction of two matrixes:

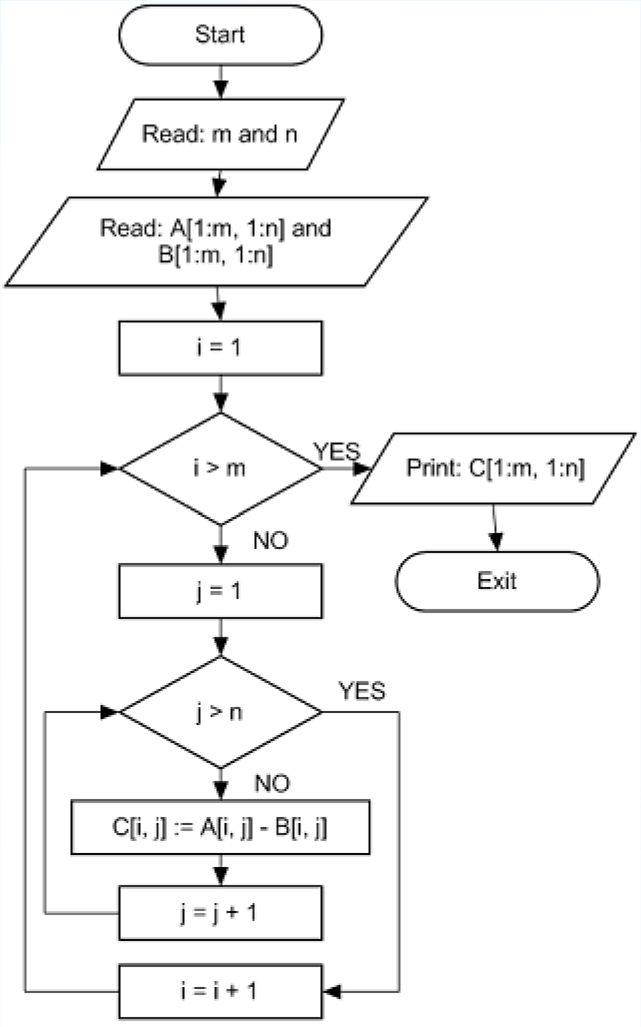
$$A - B = \begin{pmatrix} 5-3 & 10-8 & 20-5 \\ 8-2 & 6-9 & 5-3 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 15 \\ 6 & -3 & 2 \end{pmatrix}$$

Algorithm for Matrix subtraction

```
Step1: Start
Step2: Read: m and n
Step3: Read: Take inputs for Matrix A[1:m, 1:n] and Matrix B[1:m, 1:n]
Step4: Repeat for i := 1 to m by 1:
                Repeat for j := 1 to n by 1:
                        C[i, j] := A[i, j] - B[i, j]
                [End of inner for loop]
        [End of outer for loop]
Step5: Print: Matrix C
Step6: Exit.
```

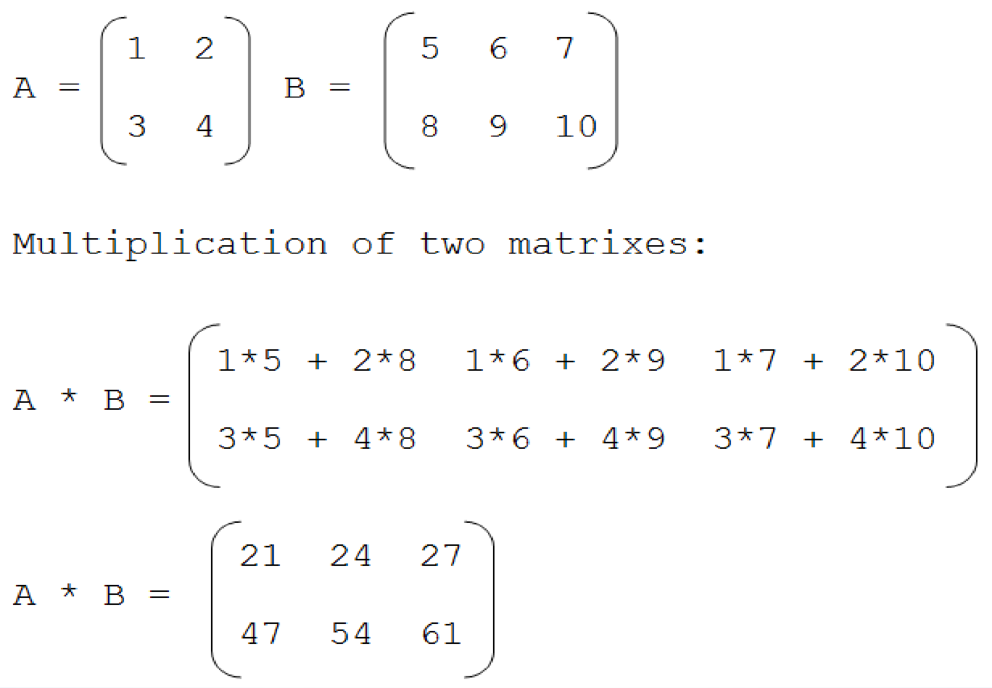
Program for matrix subtraction

Flowchart of matrix subtraction



Matrix (2D array) Multiplication

- Rule: Multiplication of two matrixes is only possible if first matrix has size m X n and other matrix has size n x r. Where m, n and r are any positive integer.
- Multiplication of two matrixes is defined as
- Program



$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

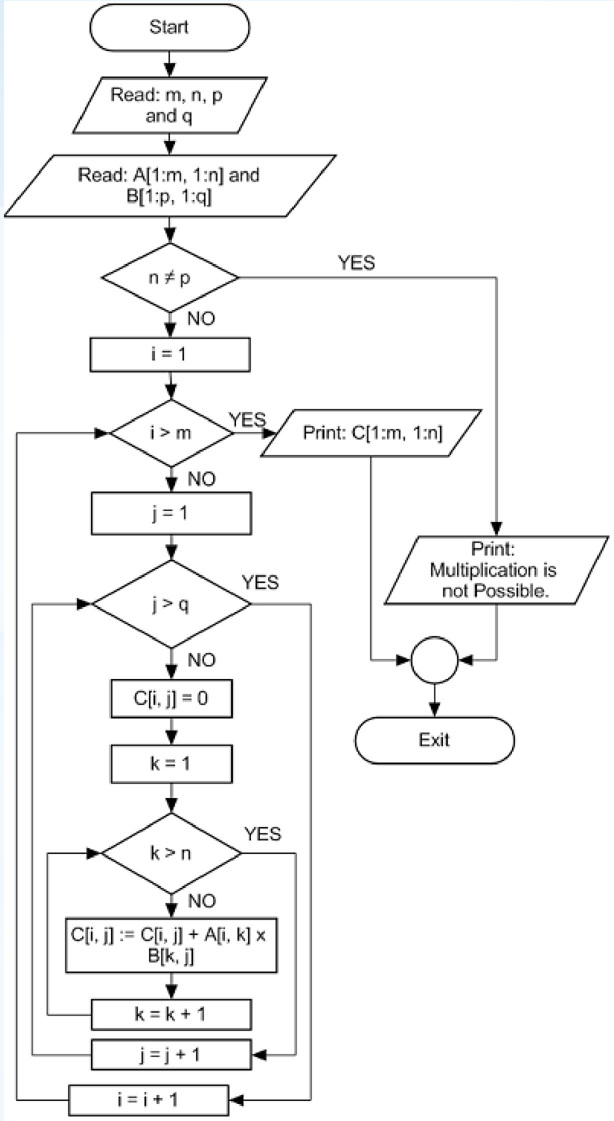
$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

Algorithm of Matrix (2D array) Multiplication

(Matrix Multiplication Algorithm) Suppose A and B are two matrices and their order are respectively m x n and p x q. i, j and k are counters. And C to store result.

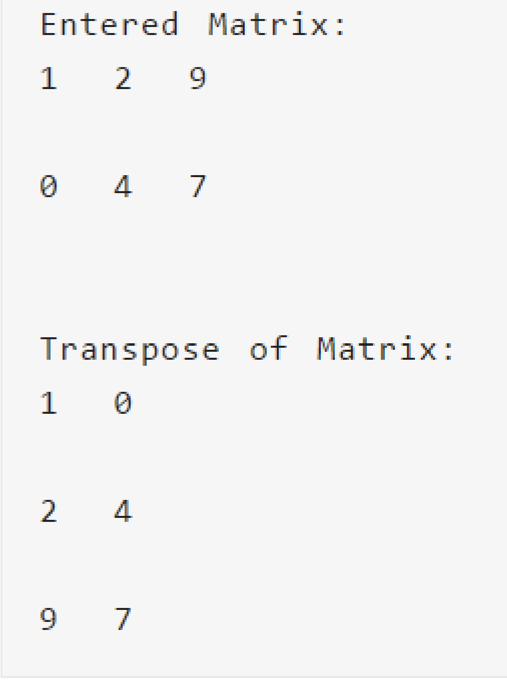
```
Step1: Start.
Step2: Read: m, n, p and q
Step3: Read: Inputs for Matrices A[1:m, 1:n] and B[1:p, 1:q].
Step4: If n ≠ p then:
        Print: Multiplication is not possible.
        Else:
           Repeat for i := 1 to m by 1:
        Repeat for j := 1 to q by 1:
        C[i, j] := 0 [Initializing]
        Repeat k := 1 to n by 1
        C[i, j] := C[i, j] + A[i, k] x B[k, j]
           [End of for loop]
        [End of for loop]
        [End of for loop]
        [End of If structure]
Step5: Print: C[1:m, 1:q]
Step6: Stop
```

Flowchart of Matrix (2D array) Multiplication



Transpose Matrix

- Transpose means, to transfer the row values into column and also the rows to column and column to row.
- The below is the example of it.
- Program for transpose

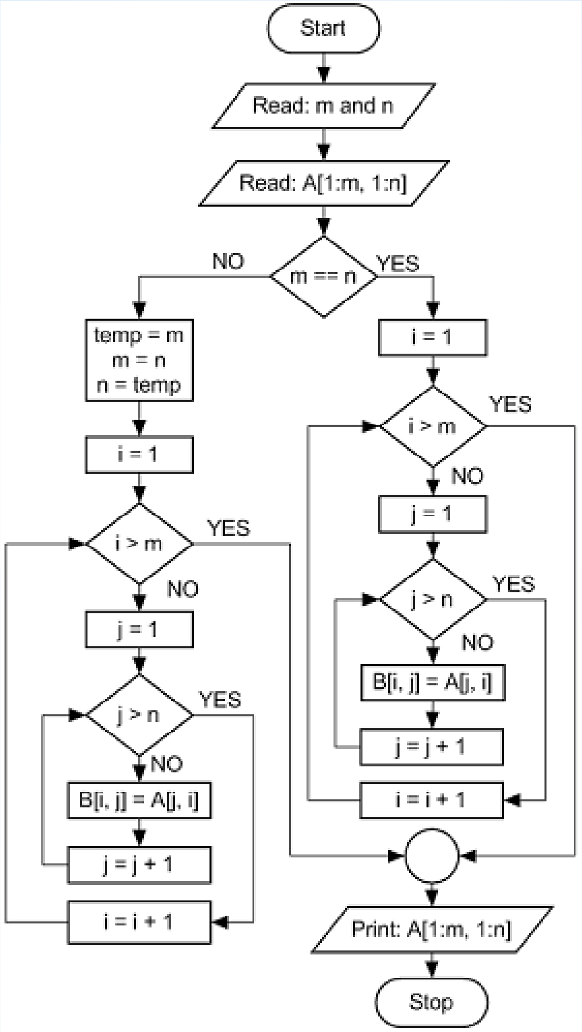


```
Entered Matrix:
1   2   9

0   4   7


Transpose of Matrix:
1   0

2   4

9   7
```

Algorithm for Transpose the matrix

```
Step1: Start.
Step2: Read: m and n
Step3: Read: Take inputs for Matrix A[1:m, 1:n].
Step4: If m == n then:
           Repeat for i = 1 to m by 1
           Repeat for j = 1 to n by 1
           B[i, j] = A[j, i]
           [End of for loop]
           [End of for loop]
           Else:
              temp = m
           m = n
           n = temp
           Repeat for i = 1 to m by 1
           Repeat for j = 1 to n by 1
           B[i, j] = A[j, i]
           [End of for loop]
           [End of for loop]
           [End of If structure]
    Step5: Print: B[1:m, 1:n]
    Step6: Exit
```

Flowchart for Transpose the matrix



Searching and Sorting on array

- Searching : To check or search entered element or data is present in an array or not.
- Two types of searching technology is there
  - Sequential (linear) Search : search element in Sequence
  - Binary Search : Divide the array in two part than perform the binary search
- Any kind of sorting and searching operation is perform by iterating loop.
- Searching Operation is done based on them index value.
- The simplest way to search for an element is to iterate through the entire array and compare each value of that array with the target element. When a match is found, we may break out of the loop.

Algorithm for sequential search

```
Step1: Start
Step2: Set J=0
Step3: Repeat steps 4 and 5 while J < N
Step4: IF LA[J] is equal ITEM THEN GOTO STEP 6
Step5: Set J = J +1
Step6: PRINT J, ITEM
Step7: Stop
```

Program for searching element from 1D array

Program for searching element from 2D array

Sorting

- Sorting is nothing but storage of data in sorted order, it can be in ascending or descending order. The term Sorting comes into picture with the term searching.

Algorithm for sorting 1D array using bubble sort

```
Step 1: start
Step 2 : take input of array size
Step 3 :declare array
Step 4 : read array element using loop
Step 5 : print original array
Step 6: for all elements of list
            if list[i] > list[i+1]
                 swap(list[i], list[i+1])
            end if
         end for
Step 7 : print sorted array
Step 8 : stop
```

Program for sorting array in ascending order

Perform below listed Program for 1D and 2D array

1) Find Average of Element using 1D array and 2D array also.
2) Declare and print array in reverse order
3) Declare one array either 2D or 1D and separate odd element and even element in two separate array named as even_array, odd_array.
4) Find Minimum element from 2D array.
5) Create array for storing 10 name and sort it into alphabetical order.
6) Write a program in C to count the frequency of each element of an array.
7) Write a program in C to find the maximum and minimum element from 1D and 2D array.
8) Write a C program to count total number of negative elements in an array.
9) Write a C program to merge two array to third array.
10) Write a C program to perform addition for two 1D array and store answer in 3rd array.