



# **2 D   A R R A Y S**

**S15-1**

# Objectives

**To learn and appreciate the following concepts**

- Programs using 2D arrays

# Session outcome

**At the end of session student will be able to**

**→ Write programs using 2D array**

# Syntax Recap

## Declaration:

```
data-type array_name[row_size][column_size];
```

## Initialization of two dimensional arrays:

```
type array-name [row size] [col size ]={list of values};
```

## Reading a Matrix:

```
int a[100][100];  
for(i=0;i<m;i++)  
{  
    for(j=0;j<n;j++)  
  
scanf("%d",&a[i][j]);  
}
```

## Display a Matrix:

```
int a[100][100];  
for(i=0;i<m;i++)  
{  
    for(j=0;j<n;j++)  
        printf("%d",a[i][j]);  
printf(" ");  
    printf("\n");  
}
```

# Trace and Norm of a Matrix

**Trace is sum of principal diagonal elements of a square matrix.**

**Norm is Square Root of sum of squares of elements of a matrix.**

```
int trace=0, sum=0,i,j,norm;
int m=3,n=3;
printf("enter elements for a
\n");
for (i=0;i<m;i++){
    for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);
}

for(i=0;i<m;i++)
    trace=trace + a[i][i];
```

```
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
        sum=sum+a[ i ][ j]*a[ i ][ j ];
}
norm=sqrt(sum);

printf(" trace is %d", trace );
printf(" norm is %d", norm );
```

# Check whether a given Matrix is Symmetric or not

```
printf("enter dimension \n");
scanf("%d %d",&m,&n);
if(m!=n)
printf("it is not a square \n");
else
{ printf("enter elements \n");
for(i=0;i<m;i++)
    for(j=0;j<n;j++)
        scanf("%d",&a[i][j]);
```

```
for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        if (a[ i ][ j ]!=a[ j ][ i ]){
            printf("\n matrix is not
                symmetric \n");
            exit(0);    }
        }
    }
printf("\n matrix is symmetric");
}
```



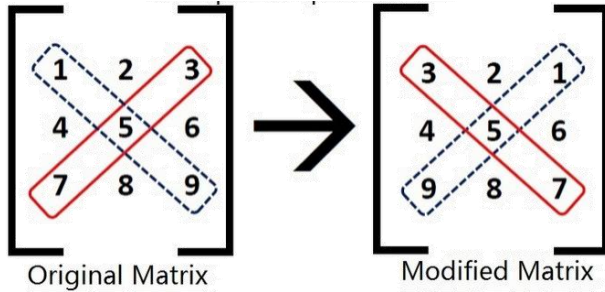
Go to posts/chat box for the link to the question **PQn. S15.1**

**submit your solution in next 2 minutes**

**The session will resume in 3 minutes**



# Exchange the elements of principal diagonal with secondary diagonal in an N dimensional Square matrix



```
int main(){
int i, j, temp, arr[4][4],n;

printf("\nEnter dimension:");
scanf("%d",&n);
printf("\nEnter elements:\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
printf("%d",arr[i][j]);
```

```
for(i=0;i<n;i++)
for(j=0;j<n;j++)
if(i==j){
temp=arr[i][j];
arr[i][j]=arr[i][n-i-1];
arr[i][n-i-1]=temp;
}
printf("\nModified Matrix:\n");
for(i=0;i<n;i++){
for(j=0;j<n;j++)
printf(" ");
printf("%d",arr[i][j]);
printf("\n");
}
return 0;}
```

# Exchange the Rows and Columns of a 'mxn' matrix

```
printf("\nEnter the cols to exchange: ");
scanf("%d %d",&c1,&c2);
/*Column exchange : c1  $\Leftrightarrow$  c2 */
for(i=0;i<m;i++) {
    temp=arr[i][c1-1];
    arr[i][c1-1]=arr[i][c2-1];
    arr[i][c2-1]=temp; }
```

```
/*read 'mxn' matrix */
printf("\nEnter the rows to exchange: ");
scanf("%d %d",&r1,&r2);
/*Row exchange r1  $\Leftrightarrow$  r2 */
for(j=0;j<n;j++) {
    temp=arr[r1-1][j];
    arr[r1-1][j]=arr[r2-1][j];
    arr[r2-1][j]=temp; }
```

# Tutorials

- Write a program to check whether the given matrix is sparse matrix or not.
- Write a program to find the sum of the elements above and below diagonal elements in a matrix.
- Write program to check the given matrix is a magic square or not

(A magic square of order  $n$  is an arrangement of  $n^2$  numbers, usually distinct integers, in a square, such that the  $n$  numbers in all rows, all columns, and both diagonals sum to the same constant. A normal magic square contains the integers from 1 to  $n^2$ . )

2	7	6	→15	
9	5	1	→15	
4	3	8	→15	
↙15	↓15	↓15	↓15	↘15

# Summary

- Declare, initialize and access 2D array
- Write programs using 2D array

# Summary of 2D arrays

- Declare, initialize and access 2D array
- Write simple programs using 2D array
- Advance programming in 2D arrays