30-05-2024

DOUBLE DIMENSIONAL ARRAYS :-

SYNTAX :- type [rows][columns]

2 for() loops are used in 2-D Arrays

	0	1	2
a[0]	1	2	3
a[1]	4	5	6

Ex:-

```
-> main ()
   {
     int a[2][3] = {{1, 2, 3}, {4, 5, 6}};
     printf ("%d %d %d %d %d %d"}, a[0][0], a[0][1],
      a[0][2], a[1][0], a[1][1], a[1][2]);
   }
```

Q:- Addition of 2 Matrices :-

```
-> main ()
   {
     int a[5][5], b[5][5], c[5][5], i, j, r1, c1;
     printf ("enter rows & cols of mat ");
     scanf ("%d %d", &r1, &c1);
     printf ("enter matrix A");

     for (i=0; i<r1; i++)
       for (j=0; j<c1; j++)
       scanf ("%d", &a[i][j]);
       printf
printf ("enter matrix B");
for (i=0; i<r1; i++)
   for (j=0; j<c1; j++)
   scanf ("%d", &b[i][j]);
for (i=0; i<r1; i++)
for (j=0; j<c1; j++]
c[i][j] = a[i][j] + b[i][j];

for( i=0; i<r1; i++, printf("\n"))
for( j=0; j<c1; j++)
printf ("%d ", c[i][j]);
}
```

Q:- WAP to check the given Matrix is Symmetry or not:-

```
-> main ()
 {
   int a[5][5], b[5][5], i, j, r, c;
   printf ("enter rows & cols of mat");
   scanf ("%d %d", &r, &c);
     if (r==c)
     {
       printf ("enter matrix A");
        for (i=0; i<r; i++)
          for(j=0; j<c; j++)
           scanf ("%d", &a[i][j]);
for (i = 0; i < r; i++)
    for (j = 0; j < c; j ++)
    b[j][i] = a[i][j];
   for (i = 0 ; i < r; i ++)
   for (j = 0; j < c; j ++)
   {
     if (a[i] [j] ! = b[i][j])
     { printf ("notSymmetry );
        exit (0);
     }
   }
   printf ("symmetry";)
  }
  else
  { printf ("Not possible ");
  }
 }
}
```

SYMMETRY :- $A = A^T$

MATRIX A :- $\begin{array}{cc} 1 & 2 \\ 2 & 5 \end{array}$ => ROWS <-> COLUMNS

SYMMETRY

Q:- Find diagonal elements :-

```
-> main ()
   {
     int a[5][5], i, j, r, c;
     printf (" enter rows & cols of mat: ");
     scanf (" %d %d ", & r, &c);
      if (r==c)
      {
        printf (" enter mat A! ");
        for (i=0; i<r; i++)
          for (j=0; j<c; j++)
        scanf ("%d", & a[i][j]);
          for (i=0; i<r; i++)
          for (j=0; j<c; j++)
            if (i==j)
            printf ("%d ", a[i][j]);
      }
      else
      printf (" Not possible ");
   }.
```

$$\begin{bmatrix} 00 & 01 & 02 \\ 10 & 11 & 12 \\ 20 & 21 & 22 \end{bmatrix}$$

1. Double dimensional array Addition of 2 matrices from user input :-

```
-> main ()
   {
     int a[100][100], b[100][100], c[100][100], i, j, r1, c1;
     printf ("Enter no of rows (between 1 and 100): ");
     Scanf ("%d", &r1);
     printf (" Enter no of columns (between 1 and 100): ");
     Scanf ("%d", &c1);

     printf ("\n Enter matrix A: \n");
     for (i=0; i<r1; i++)
        for (j=0; j<c1; j++)
        scanf ("%d", &a[i][j]);

        printf ("\n Enter matrix B: \n");
        for (i=0; i<r1; i++)
          for (j=0; j<c1; j++)
           Scanf ("%d", &b[i][j]);

        for(i=0; i<r1; i++)
        for (j=0; j<c1; j++)
        c[i][j]= a[i][j]+ b[i][j];

        printf ("\n Sum of two matrices: \n");
        for (i=0; i<r1; ++i)
        for (j=0; j<c1; ++j)
          { printf ("%d  ", c[i][j]);
            if (j= C1 - 1)
            {
              printf ("\n");
   }      }  }
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

$$= \begin{bmatrix} 1\times 4+2\times 2 & 1\times 3+2\times 1 \\ 3\times 4+4\times 2 & 3\times 3+4\times 1 \end{bmatrix}$$

OUTPUT:-

```
Enter number of rows [between 1 ~~and~~ and ~~to~~ 100]: 3
Enter number of columns [between 1 and 100]: 3

Enter matrix A:

1  2  3
4  5  6
7  8  9

Enter matrix B:

10  11  12
13  14  15
16  17  18

Sum of two matrices:

11  13  15
17  19  21
23  25  27
```