

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]);
}
```

a:- 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<n1; i++)
```

```
for (j=0; j<c1; j++)
```

```
scanf ("%d", &b[i][j]);
```

```
for (i=0; i<n1; i++)
```

```
for (j=0; j<c1; j++)
```

```
c[i][j] = a[i][j] + b[i][j];
```

```
for (i=0; i<n1; 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, n, c;
```

```
printf ("enter rows & cols of mat");
```

```
scanf ("%d %d", &n, &c);
```

```
if (n == c)
```

```
{
```

```
printf ("enter matrix A");
```

```
for (i=0; i<n; i++)
```

```
for (j=0; j<c; j++)
```

```
scanf ("%d", &a[i][j]);
```



```
for (i = 0; i < n; i++)
```

```
for (j = 0; j < c; j++)
```

```
b[j][i] = a[i][j];
```

```
for (i = 0; i < n; i++)
```

```
for (j = 0; j < c; j++)
```

```
{
```

```
if (a[i][j] != b[j][i])
```

```
{ printf ("Not Symmetry");
```

```
exit (0);
```

```
}
```

```
}
```

```
printf ("Symmetry")
```

```
}
```

```
else
```

```
{
```

```
printf ("Not possible");
```

```
}
```

```
}
```

```
}
```

SYMMETRY :- $A = A^T$

MATRIX A :- $\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} \Rightarrow \text{Rows} \leftrightarrow \text{Columns}$

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");

}

0	1	0
1	1	1
2	0	2

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} 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

2. Sum of upper triangle & lower triangle in a matrix using double dimensional array: +

→ main ()

```
{  
    int a[100][100], b[100][100], c[100][100], i, j, r1, c1;  
    int upper-sum = 0;  
    int lower-sum = 0;  
    printf ("Enter number of rows (between 1 and 100): ");  
    scanf ("%d", &r1);  
    printf ("Enter number of columns (between 1 and 100): ");  
    scanf ("%d", &c1);  
    printf ("\n Enter matrix A: \n");
```

// To calculate sum of upper triangle

```
for (i=0; i<r1; i++)
```

```
    for (j=0; j<c1; j++)
```

```
        scanf ("%d", &a[i][j]);
```

// To calculate sum of upper triangle

```
for (i=0; i<r1; i++)
```

```
    for (j=0; j<c1; j++)
```

```
{
```

```
    if (i <= j)
```

```
        upper-sum += a[i][j];
```

```
}
```

```
printf ("\n Sum of upper triangular matrix  
is : %d \n", upper-sum);
```


// To calculate sum of lower triangle

```
for (i = 0; i < n1; i++)
```

```
for (j = 0; j < c1; j++)
```

```
{
```

```
    if (j <= i)
```

```
        lower_sum += a[i][j];
```

```
}
```

```
printf ("Sum of lower triangular matrix  
is: '%d'", lower_sum);
```

```
}
```

OUTPUT:-

Enter number of rows [between 1 and 100]: 3

Enter number of columns [between 1 and 100]: 3

Enter matrix A :

1 2 3

4 5 6

7 8 9

1 2 3

5 6 26

9

Sum of upper triangular matrix is = 26

Sum of lower triangular matrix is = 34

4 5 = 34

7 8 9

3. Program to find the sum of diagonal elements in a matrix;

-> main ()

{

int a[100][100], b[100][100], i, j, r, c, sum;

printf ("Enter rows (between 1 and 100):");

scanf ("%d", &r);

printf ("Enter columns (between 1 and 100): ");

scanf ("%d", &c);

if (r == c)

{

printf ("Enter matrix A: (%d)",

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)

sum += a[i][j];

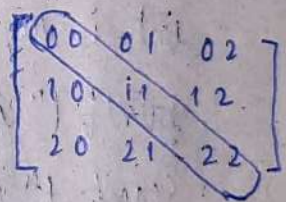
printf ("Sum of the diagonal elements in a
matrix is: %d (%d)", sum);

}

else

printf ("Not possible");

}



00	01	02
10	11	12
20	21	22

00 + 11 + 22

01-06-2024

→ main()

```
{  
    char a[5][20] = {"a", "b", "c", "d", "e"};  
    int i;  
    for (i=0; i<5; i++)  
        printf ("%s\n", a[i]);  
}
```

Q:- Reading & Printing of n-strings:-

→ main()

```
{  
    char a[10][20];  
    int i, n;  
    printf ("enter n");  
    scanf ("%d", &n);  
    printf ("enter %d strings", n);  
    for (i=0; i<n; i++)  
        scanf ("%s", a[i]);  
    for (i=0; i<n; i++)  
        printf ("%s\n", a[i]);  
}
```


Q:- search a string :-

-> main ()

{

char a[20][20], s[20];

int i, m;

printf ("enter m");

scanf ("%d", &m);

printf ("enter %d strings", m);

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

scanf ("%s", a[i]);

printf ("enter search string");

scanf ("%s", t);

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

{

if (strcmp(a[i], t) == 0)

printf ("string is found at place %d", i+1);

exit(0);

}

printf ("not found");

}

Q:- WAP to sort n-strings in ascending order:-

→ main ()

```
{
    char a[20][20], t[20];
    int i, j, n;
    printf ("enter n");
    scanf ("%d", &n);
    printf ("enter %d string", n);
    for (i=0; i<n; i++)
        scanf ("%s", a[i]);
    for (i=0; i<n-1; i++)
    {
        for (j=0; j<n-i-1; j++)
        {
            if (strcmp(a[j], a[j+1]) > 0)
            {
                strcpy(t, a[j]);
                strcpy(a[j], a[j+1]);
                strcpy(a[j+1], t);
            }
        }
    }
    for (i=0; i<n; i++)
        printf ("%s \n", a[i]);
}
```