

## Assignment - 16 A Job Ready Bootcamp in C++, DSA and IOT MySirG

### Multi-Dimensional Array in C Language

1. Write a program to calculate the sum of two matrices each of order 3x3.

**Program -**

```
#include<stdio.h>

int main()
{
    int a[3][3] , b[3][3] , i , j;

    printf("\nEnter first matrix elements:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        printf("Enter 3 elements in row-%d\n",i+1);
        for(j = 0 ; j < 3 ; j++)
        {
            scanf("%d",&a[i][j]);
        }
    }

    printf("\nEnter second matrix elements:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        printf("Enter 3 elements in row-%d\n",i+1);
        for(j = 0 ; j < 3 ; j++)
        {
            scanf("%d",&b[i][j]);
        }
    }

    printf("\nResultant matrix is:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        for(j = 0 ; j < 3 ; j++)
        {
            printf("%d  ",a[i][j]+b[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

**Output -**

```
Enter first matrix elements:-
Enter 3 elements in row-1
1 2 3
Enter 3 elements in row-2
```

8 1 3  
Enter 3 elements in row-3  
0 7 2  
Enter second matrix elements:-  
Enter 3 elements in row-1  
1 1 1  
Enter 3 elements in row-2  
0 0 1  
Enter 3 elements in row-3  
0 0 3  
Resultant Matrix is:-  
2 3 4  
8 1 4  
0 7 5

---

**2. Write a program to calculate the product of two matrices each of order 3x3.**

**Program -**

```
#include<stdio.h>

int main()
{
    int a[3][3] , b[3][3] , c[3][3], i , j , k , sum;

    printf("Enter first matrix elements:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        printf("Enter 3 elements in row-%d\n",i+1);
        for(j = 0 ; j < 3 ; j++)
        {
            scanf("%d",&a[i][j]);
        }
    }

    printf("\nEnter second matrix elements:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        printf("Enter 3 elements in row-%d\n",i+1);
        for(j = 0 ; j < 3 ; j++)
        {
            scanf("%d",&b[i][j]);
        }
    }

    printf("\nProduct of both matrices is:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        for(j = 0 ; j < 3 ; j++)
        {
            sum = 0;
```

```

        for(k = 0 ; k < 3 ; k++)
        {
            sum += a[i][k] * b[k][j];
        }
        c[i][j] = sum;
        printf(" %d  ",c[i][j]);
    }
    printf("\n");
}
return 0;
}

```

### Output -

Enter first matrix elements:-

Enter 3 elements in row-1

4 5 3

Enter 3 elements in row-2

1 2 3

Enter 3 elements in row-3

0 9 0

Enter second matrix elements:-

Enter 3 elements in row-1

8 8 2

Enter 3 elements in row-2

8 9 3

Enter 3 elements in row-3

5 5 4

Product of both matrices is:-

87 92 35

39 41 20

72 81 27

### 3. Write a program in C to find the transpose of a given matrix.

#### Program -

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int rows , cols , i , j;
```

```
    printf("Enter number of rows and columns:-\n");
```

```
    printf("Rows : ");
```

```
    scanf("%d",&rows);
```

```
    printf("Columns : ");
```

```
    scanf("%d",&cols);
```

```
    int arr[rows][cols];
```

```
    printf("Enter a %d x %d matrix:-\n",rows,cols);
```

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

```

{
    printf("Enter %d elements in row - %d elements\n",cols,i+1);
    for(j = 0 ; j < cols ; j++)
    {
        scanf("%d",&arr[i][j]);
    }
}

printf("Entered matrix is:-\n");
for(i = 0 ; i < rows ; i++)
{
    for(j = 0 ; j < cols ; j++)
    {
        printf(" %d  ",arr[i][j]);
    }
    printf("\n");
}

printf("It's transpose is:-\n");
for(i = 0 ; i < cols ; i++)
{
    for(j = 0 ; j < rows ; j++)
    {
        printf(" %d  ",arr[j][i]);
    }
    printf("\n");
}
return 0;
}

```

### Output -

```

Enter number of rows and columns:-
Rows : 3
Columns : 2
Enter a 3 x 2 matrix:-
Enter 2 elements in row - 1 elements
1 2
Enter 2 elements in row - 2 elements
5 6
Enter 2 elements in row - 3 elements
9 0
Entered matrix is:-
1 2
5 6
9 0
It's transpose is:-
1 5 9
2 6 0

```

---

#### 4. Write a program in C to find the sum of right diagonals of a matrix.

Program -

```
#include<stdio.h>

int main()
{
    int rows , cols , i , j;

    printf("Enter a square matrix:-\n");
    printf("Rows : ");
    scanf("%d",&rows);
    printf("Columns : ");
    scanf("%d",&cols);

    if(rows != cols)
    {
        printf("Not a square matrix");
        return 0;
    }

    int arr[rows][cols];
    printf("Enter a %d x %d matrix:-\n",rows,cols);
    for(i = 0 ; i < rows ; i++)
    {
        printf("Enter %d elements in row - %d\n",cols,i+1);
        for(j = 0 ; j < cols ; j++)
        {
            scanf("%d",&arr[i][j]);
        }
    }

    printf("Entered matrix is:-\n");
    for(i = 0 ; i < rows ; i++)
    {
        for(j = 0 ; j < cols ; j++)
        {
            printf(" %d  ",arr[i][j]);
        }
        printf("\n");
    }

    int sum = 0;
    for(i = 0 ; i < rows ; i++)
    {
        for(j = 0 ; j < cols ; j++)
        {
            if(j == (rows - 1) - i)
            {
                sum += arr[i][j];
            }
        }
    }
}
```

```

    }
}

printf("Sum of it's right diagonal elements = %d",sum);

printf("\n");
return 0;
}

```

### Output -

```

Enter a square matrix:-
Rows : 4
Columns : 4
Enter a 4 x 4 matrix:-
Enter 4 elements in row - 1
1 2 3 4
Enter 4 elements in row - 2
5 6 7 8
Enter 4 elements in row - 3
1 2 3 2
Enter 4 elements in row - 4
0 0 0 2
Entered matrix is:-
1 2 3 4
5 6 7 8
1 2 3 2
0 0 0 2
Sum of it's right diagonal elements = 13

```

---

### 5. Write a program in C to find the sum of left diagonals of a matrix.

#### Program -

```

#include<stdio.h>

int main()
{
    int rows , cols , i , j;
    printf("Enter a square matrix:-\n");
    printf("Rows : ");
    scanf("%d",&rows);
    printf("Columns : ");
    scanf("%d",&cols);

    if(rows != cols)
    {
        printf("Not a square matrix");
        return 0;
    }
}

```

```

int arr[rows][cols];
printf("Enter a %d x %d matrix:-\n",rows,cols);
for(i = 0 ; i < rows ; i++)
{
    printf("Enter %d elements in row - %d\n",cols,i+1);
    for(j = 0 ; j < cols ; j++)
    {
        scanf("%d",&arr[i][j]);
    }
}

printf("Entered matrix is:-\n");
for(i = 0 ; i < rows ; i++)
{
    for(j = 0 ; j < cols ; j++)
    {
        printf(" %d  ",arr[i][j]);
    }
    printf("\n");
}

int sum = 0;
for(i = 0 ; i < rows ; i++)
{
    for(j = 0 ; j < cols ; j++)
    {
        if(j == i)
        {
            sum += arr[i][j];
        }
    }
}

printf("Sum of it's left diagonal elements = %d",sum);

printf("\n");
return 0;
}

```

### Output -

Enter a square matrix:-

Rows : 3

Columns : 3

Enter a 3 x 3 matrix:-

Enter 3 elements in row - 1

1 2 3

Enter 3 elements in row - 2

0 9 0

Enter 3 elements in row - 3

0 0 1

Entered matrix is:-

```
1  2  3
0  9  0
0  0  1
```

Sum of it's left diagonal elements = 11

---

**6. Write a program in C to find the sum of rows and columns of a Matrix.**

**Program -**

```
#include<stdio.h>
```

```
int main()
{
    int rows , cols , i , j , sum_rows , sum_cols;

    printf("Enter a matrix:-\n");
    printf("Enter number of Rows : ");
    scanf("%d",&rows);
    printf("Enter number of Columns : ");
    scanf("%d",&cols);

    int arr[rows][cols];
    printf("Enter a %d x %d matrix:-\n",rows,cols);
    for(i = 0 ; i < rows ; i++)
    {
        printf("Enter %d elements in row - %d\n",cols,i+1);
        for(j = 0 ; j < cols ; j++)
        {
            scanf("%d",&arr[i][j]);
        }
    }

    printf("Entered matrix is:-\n");
    for(i = 0 ; i < rows ; i++)
    {
        for(j = 0 ; j < cols ; j++)
        {
            printf(" %d  ",arr[i][j]);
        }
        printf("\n");
    }

    for(i = 0 ; i < rows ; i++)
    {
        sum_rows = 0;
        for(j = 0 ; j < cols ; j++)
        {
            sum_rows += arr[i][j];
        }
        printf("Sum of elements of row-%d : %d\n",i+1,sum_rows);
    }
}
```



```

    }

    for(i = 0 ; i < cols ; i++)
    {
        sum_cols = 0;
        for(j = 0 ; j < rows ; j++)
        {
            sum_cols += arr[j][i];
        }
        printf("Sum of elements of column-%d : %d\n",i+1,sum_cols);
    }

    return 0;
}

```

### Output -

```

Enter a matrix:-
Enter number of Rows : 2
Enter number of Columns : 3
Enter a 2 x 3 matrix:-
Enter 3 elements in row - 1
3 4 5
Enter 3 elements in row - 2
6 7 3
Entered matrix is:-
3 4 5
6 7 3
Sum of elements of row-1 : 12
Sum of elements of row-2 : 16
Sum of elements of column-1 : 9
Sum of elements of column-2 : 11
Sum of elements of column-3 : 8

```

---

**7. Write a program in C to print or display the lower triangular of a given matrix.**

### Program -

```

#include<stdio.h>

int main()
{
    int arr[3][3] = {{1,0,0},{5,6,0},{5,4,4}};
    int i , j;

    printf("Matrix is:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        for(j = 0 ; j < 3 ; j++)
        {
            printf("%d  ",arr[i][j]);

```

```

    }
    printf("\n");
}

printf("Lower triangular matrix is:-\n");
for(i = 0 ; i < 3 ; i++)
{
    for(j = 0 ; j < 3 ; j++)
    {
        if(i < j)
            printf("    ");
        else
            printf("%d  ",arr[i][j]);
    }
    printf("\n");
}

return 0;
}

```

#### Output -

Matrix is:-

1 0 0

5 6 0

5 4 4

Lower triangular matrix is:-

1

5 6

5 4 4

---

**8. Write a program in C to print or display an upper triangular matrix.**

#### Program -

```
#include<stdio.h>
```

```

int main()
{
    int arr[3][3] = {{1,2,3},{0,8,6},{0,0,4}};
    int i , j;

    printf("Matrix is:-\n");
    for(i = 0 ; i < 3 ; i++)
    {
        for(j = 0 ; j < 3 ; j++)
        {
            printf("%d  ",arr[i][j]);
        }
        printf("\n");
    }
}

```

```

printf("Upper triangular matrix is:-\n");
for(i = 0 ; i < 3 ; i++)
{
    for(j = 0 ; j < 3 ; j++)
    {
        if(i > j)
            printf("  ");
        else
            printf("%d  ",arr[i][j]);
    }
    printf("\n");
}

return 0;
}

```

### Output -

Matrix is:-

1 2 3

0 8 6

0 0 4

Upper triangular matrix is:-

1 2 3

8 6

4

**9. Write a program in C to accept a matrix and determine whether it is a sparse matrix.**

### Program -

```
#include<stdio.h>
```

```

int main()
{
    int rows , cols , i , j , count = 0;

    printf("Enter number of Rows : ");
    scanf("%d",&rows);
    printf("Enter number of Columns : ");
    scanf("%d",&cols);

    int arr[rows][cols];
    printf("Enter a %d x %d matrix:-\n",rows,cols);
    for(i = 0 ; i < rows ; i++)
    {
        printf("Enter %d elements in row - %d\n",cols,i+1);
        for(j = 0 ; j < cols ; j++)
        {
            scanf("%d",&arr[i][j]);
            if(arr[i][j] == 0)
                count++;
        }
    }
}

```

```

    }
}

printf("Matrix is:-\n");
for(i = 0 ; i < rows ; i++)
{
    for(j = 0 ; j < cols ; j++)
    {
        printf(" %d  ",arr[i][j]);
    }
    printf("\n");
}

if(count > (rows*cols)/2)
{
    printf("It is a sparse matrix.\n");
    printf("There are %d zeros. This count is more than half of
matrix size.",count);
}
else
    printf("Not a sparse matrix");

return 0;
}

```

### Output -

```

Enter number of Rows : 3
Enter number of Columns : 3
Enter a 3 x 3 matrix:-
Enter 3 elements in row - 1
1 0 0
Enter 3 elements in row - 2
0 8 4
Enter 3 elements in row - 3
0 0 0
Matrix is:-
1 0 0
0 8 4
0 0 0
It is a sparse matrix.
There are 6 zeros. This count is more than half of matrix size.

```

---

**10. Write a program in C to find the row with maximum number of 1s.**

### Program -

```

#include<stdio.h>

int main()
{
    int rows , cols , i , j , count , max = -1;

```

```

printf("Enter number of Rows : ");
scanf("%d",&rows);
printf("Enter number of Columns : ");
scanf("%d",&cols);

int arr[rows][cols];
printf("Enter a %d x %d matrix:-\n",rows,cols);
for(i = 0 ; i < rows ; i++)
{
    printf("Enter %d elements in row - %d\n",cols,i+1);
    for(j = 0 ; j < cols ; j++)
    {
        scanf("%d",&arr[i][j]);
    }
}

printf("Matrix is:-\n");
for(i = 0 ; i < rows ; i++)
{
    for(j = 0 ; j < cols ; j++)
    {
        printf(" %d  ",arr[i][j]);
    }
    printf("\n");
}

int ones[rows];
for(i = 0 ; i < rows ; i++)
{
    count = 0;
    for(j = 0 ; j < cols ; j++)
    {
        if(arr[i][j] == 1)
            count++;
    }
    ones[i] = count;
}

for(i = 0 ; i < rows ; i++)
{
    if(max < ones[i])
        max = ones[i];
}

for(i = 0 ; i < rows ; i++)
{
    if(ones[i] == max)
    {
        printf("Maximum number of 1s are there in row -
%d\n",i+1);
    }
}

```

```
        }  
    }  
  
    return 0;  
}
```

### Output -

Enter number of Rows : 3

Enter number of Columns : 4

Enter a 3 x 4 matrix:-

Enter 4 elements in row - 1

1 1 1 9

Enter 4 elements in row - 2

0 0 1 1

Enter 4 elements in row - 3

1 1 1 1

Matrix is:-

1 1 1 9

0 0 1 1

1 1 1 1

Maximum number of 1s are there in row - 3