

**PROGRAM TITLE:Create a Spiral matrix(clockwise) of length
r x c.**

THEORY:A matrix which stores its elements in a spiral manner is known as a Spiral matrix. In our Program, we create a clockwise Spiral matrix with the inputs from the user.

PROGRAM ALGORITHM:

```
Algo_CreateSpiral(mat,r,c)
{
    while(number of inputs less than r*c)
    {
        if(count even)
            Put the elements in the upper right boundary of the matrix
        else
            Put the elements in the lower left boundary of the matrix
    }
}
```

PROGRAM CODE:

```
/*C Program to Create a Spiral Matrix*/
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i,j,r,c,ch=0,e=0,ur,uc,lr=0,lc=0;
    printf("\n\tEnter size ::");
    scanf("%d %d",&r,&c);
    ur=r-1;
    uc=c-1;

    /*Allocate memory space for the matrix*/
    int **mat=(int **)malloc(r*sizeof(int *));
    if(!mat)
    {
        printf("\n\tAllocation failed");
        return 2;
    }
    for(i=0;i<r;i++)
    {
        mat[i]=(int *)malloc(c*sizeof(int));
        if(!mat[i])
        {
            printf("\n\tAllocation failed");
            return 1;
        }
    }
    i=0;

    /*Loop to create the Matrix as Spiral*/
    while(e<r*c)
```

```

{
    if(ch%2==0)
    {
        /*Put the elements in the upper right boundary of the matrix
*/
        for(j=lc;j<=uc;j++)
        {
            printf("\n\tEnter element at %d %d::\t",i,j);
            scanf("%d",&mat[i][j]);e++;
        }
        lr++;j--;
        for(i=lr;i<=ur;i++)
        {
            printf("\n\tEnter element at %d %d::\t",i,j);
            scanf("%d",&mat[i][j]);e++;
        }
        uc--;i--;
    }
    else
    {
        /*Put the elements in the lower left boundary of the matrix */
        for(j=uc;j>=lc;j--)
        {
            printf("\n\tEnter element at %d %d::\t",i,j);
            scanf("%d",&mat[i][j]);e++;
        }
        ur--;j++;
        for(i=ur;i>=lr;i--)
        {
            printf("\n\tEnter element at %d %d::\t",i,j);
            scanf("%d",&mat[i][j]);e++;
        }
        lc++;i++;
    }
    ch++;
}

/*Print the matrix in its final form*/
printf("\n\tThe Matrix is ::\n");
for(i=0;i<r;i++)
{
    for(j=0;j<c;j++)
    {
        printf("\t%d",mat[i][j]);
    }
    printf("\n");
}
return 0;
}

```

OUTPUT:

Enter size ::4 3

```
Enter element at 0 0::      1
Enter element at 0 1::      2
Enter element at 0 2::      3
Enter element at 1 2::      4
Enter element at 2 2::      5
Enter element at 3 2::      6
Enter element at 3 1::      7
Enter element at 3 0::      8
Enter element at 2 0::      9
Enter element at 1 0::     10
Enter element at 1 1::     11
Enter element at 2 1::     12
```

The Matrix is ::

```
1      2      3
10     11     4
9      12     5
8      7      6
```

DISCUSSION:

The complexity of the Program is $O(n^2)$. The Program also works when row number is not equal to column number. The Program shows the position where the element is being inserted.