**6. Topic – Printing Spiral Matrix using C program**

* **Problem Statement**

Create an n × n square matrix dynamically taking n as an input from the user. Fill the matrix with integers 1, 2, ..., n^2 in a spiral order starting from the left top corner of the matrix. Print the matrix.

**Input example:** /\* Here user will give the order of the square matrix\*/

Enter the order of the matrix

5

**Output example:**

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

* **Proposed C Code**

**/\* ---------- spiralmatrix.c--------------- \*/**

#include<stdio.h>

#include<stdlib.h>

int main()

{ int n;

printf("Enter the order of the matrix\n");

scanf("%d",&n);

/\* A 2D array is created for the nxn matrix using dynamically \*/

int \*\*arr = (int\*\*)malloc(n\*sizeof(int\*));

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

{

arr[i] = (int\*)malloc(n\*sizeof(int));

}

/\* Two top and bottom variables and two left and right variables declared and initialized globally for moving in the matrix

top to bottom,bottom to top,left to right and right to left \*/

int top1=0,top2=0,bottom1=n-1,bottom2=n-1;

int left1=0,left2=0,right1=n-1,right2=n-1;

int c=1;

/\* The c variable will enter the values from 1 to n^2 \*/

while ( c < n\*n )

{

/\* For loop for entering values from left to right using global variables left1 and right1 \*/

/\* c is incremented in every for loop \*/

for ( int i = left1 ; i < right1 ; i++ )

{

arr[left1][i]=c;

c++;

if(c==n\*n)

break;

}

/\* After every step left1 is increased and right1 is decreased \*/

left1++;

right1--;

/\* For loop for entering values from top to bottom using global variables top1 and bottom1 \*/

for ( int i = top1 ; i < bottom1 ; i++ )

{

arr[i][bottom1]=c;

c++;

if(c==n\*n)

break;

}

/\* After every step top1 is increased and bottom1 is decreased \*/

top1++;

bottom1--;

/\* For loop for entering values from right to left using global variables right2 and left2 \*/

for ( int i = right2 ; i > left2 ; i-- )

{

arr[right2][i]=c;

c++;

if(c==n\*n)

break;

}

/\* After every step left2 is increased and right2 is decreased \*/

left2++;

right2--;

/\* For loop for entering values from bottom to top using global variables bottom2 and top2 \*/

for ( int i = bottom2 ; i > top2 ; i-- )

{

arr[i][top2]=c;

c++;

if(c==n\*n)

break;

}

/\* After every step top2 is increased and bottom2 is decreased \*/

top2++;

bottom2--;

}

/\* If n is odd we have to separately print n^2 after the loop as for odd n 4 for loops will be exexuted and c with value n^2 will not enter the while loop for odd n,for even n, no problem will arise \*/

if(n%2==1)

{

arr[n/2][n/2]=n\*n;

}

/\* The nxn matrix is printed with the numbers 1 to n^2 in a spiral order\*/

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

{

for(int j=0;j<n;j++)

{

printf("%3d ",arr[i][j]);

}

printf("\n");

}

return 0;

}

**/\*------------------------------------------------------------------------------------------------------------------------- \*/**

* **Conclusion**

**The proposed algorithm has a runtime of O(n^2) where n is generally input order of the square matrix under consideration.**

* **Limitations : If we locally declare any array then its size should be less than 10^5 which we have to consider so the size of the matrix in this program is considered to be less than 10^5.**
* **Assumptions: Here we are considering the strings to be positive integers of ‘int’ datatype so the input size is withen the positive intger range.**