FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

11th October 2021

CODE: (code pasted in this format for readability)

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

void display (int a[10][10], int r, int c)
{
    for (int i=0; i<r; i++)
    {
        for (int j=0; j<c; j++)
         printf("\t%d", a[i][j]);
        printf("\n");
    }
}

void display_helical (int a[10][10], int r, int c)
{
    int i, r_index=0, c_index=0;            //c_index and r_index chosen to access indexes in spiral order
    while (r_index<r && c_index<c)          //to print the array in spiral form within bounds
    {
        for (i=c_index; i<c; i++)           //prints the first row of the 2d array in the first step and the reduced 2d array in subsequent steps
            printf("%d ", a[r_index][i]);
        r_index++;                          //to increment the row index so as to start with next row (2d array reduction)
        for (i=r_index; i<r; i++)           //prints the last column of the 2d array in the first step and the reduced 2d array in subsequent steps
            printf("%d ", a[i][c-1]);
        c--;                                //to decrement the no of columns so as to start with previous column (2d array reduction)
        if (r_index<r)                      //for printing the remaining row (last row in first iteration)
        {
            for (i=c-1; i>=c_index; i--)
                printf("%d ", a[r-1][i]);
            r--;                            //decrementing no of rows to reduce 2d array
        }
        if (c_index<c)                      //for printing the remaining column (first column in first iteration)
        {
```

```
            for (i=r-1; i>=r_index; i--)
                printf("%d ", a[i][c_index]);
            c_index++;                      //incrementing no of columns to reduce 2d
array
        }
    }
    printf("\n");
}

void display_helical_anti (int a[10][10], int r, int c)
{
    int i, r_index=0, c_index=0;
    while (r_index<r && c_index<c)
    {
        for (i=c-1; i>=c_index; i--)
            printf("%d ", a[r_index][i]);
        r_index++;
        for (i=r_index; i<r; i++)
            printf("%d ", a[i][0]);
        c_index++;
        if (r_index<r)
        {
            for (i=c_index; i<c; i++)
                printf("%d ", a[r-1][i]);
            r--;
        }
        if (c_index<c)
        {
            for (i=r-1; i>=r_index; i--)
                printf("%d ", a[i][c-1]);
            c--;
        }
    }
    printf("\n");
}

int main()
{
    int r, c, ch;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n\n");
    A:
    printf("Enter the number of rows and columns for the 2d matrix (max 10 each):\n");
    scanf("%d%d", &r, &c);
    int a[10][10];
    printf("Enter the elements (row major):\n");
    for (int i=0; i<r; i++)
    {
        for (int j=0; j<c; j++)
         scanf("%d", &a[i][j]);
    }
    printf("\nArray elements:\n");
    display(a,r,c);
    while (1)
    {
```

```
        B:
        printf("\n\nMENU\n1. Print in helical order.\n2. Print in reverse helical
order.\n3. Enter another array.\n4. Exit.\n");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: printf("\nArray elements printed in helical order:\n");
                    display_helical(a,r,c);
                    break;
            case 2: printf("\nArray elements printed in reverse helical order:\n");
                    display_helical_anti(a,r,c);
                    break;
            case 3: goto A;
            case 4: exit(0);
            default: printf("\nWrong choice! Enter again...\n");
                     goto B;

        }
    }
    return 0;
}
```

OUTPUT:

```
Enter the number of rows and columns for the 2d matrix (max 10 each):
3 4
Enter the elements (row major):
1 2 3 4
5 6 7 8
9 10 11 12

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

```
MENU
1. Print in helical order.
2. Print in reverse helical order.
3. Enter another array.
4. Exit.
1

Array elements printed in helical order:
1 2 3 4 8 12 11 10 9 5 6 7
```

```
MENU
1. Print in helical order.
2. Print in reverse helical order.
3. Enter another array.
4. Exit.
2

Array elements printed in reverse helical order:
4 3 2 1 5 9 10 11 12 8 7 6
```