DataStructure Lab-I

Instruction:

a) You are supposed to allocate the array or matrix using malloc functions.
b) You must ensure that your solution defends any test-cases. For instance, 1x1 matrix.
c) Input should be given by random function and the code snippet of a single dimensional array is given below-

```
#include<stdlib.h>
#include<time.h>
srand(time(0));
for(i=0;i<n;i++)
    a[i]=(rand()%(upper_limit-lower_limit +1))+lower_limit;
```

Problems:

1. Write a program to rotate a matrix in 90 degree angle.
2. Write a program to print a matrix in spiral order.
3. Given an array arr[] of positive integers of size N. Reverse every sub-array of K group elements.

National Institute of Technology Silchar

LABORATORY EXERCISE BOOK

Name: Subhojit Ghimire

Scholar ID: 1912160 **Sec: CSE B**

Subject: CS211

CSE 2023

INDEX

S. No.	Name of the Programme
1.	Write a program to rotate a matrix in 90 degree angle.
2.	Write a program to print a matrix in spiral order.
3.	Given an array arr[] of positive integers of size N. Reverse every sub-array of K group elements.

Signature of Instructor

```c
//Q.1. Write a program to rotate a matrix in 90 degree angle.

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


int main()
{
    int **mat, **mat1, row, col, i, j, k, temp;

    printf ("Enter the number of rows for your matrix: ");
    scanf ("%d", &row);
    mat=(int **) malloc (row * sizeof(int));
    mat1=(int **) malloc (row * sizeof(int));

    printf("Enter the number of columns for your matrix: ");
    scanf ("%d", &col);
    for (i=0; i<row; ++i)
    {
        mat [i] = (int *) malloc (col * sizeof(int));
        mat1 [i] = (int *) malloc (col * sizeof(int));
    }


    printf ("Randomly Entering values in matrix... Please Wait...");

    srand (time(0));
    for (i=0; i<row; ++i)
        for(j=0; j<col; ++j)
        {
            mat[i][j] = (rand() % (100));
        }
    printf("\nVlaues successfully stored in the matrix... This is the matrix created \n");

    for (i=0; i<row; ++i)
    {
        for (j=0 ; j<row; ++j)
        {
            printf("%d\t", mat[i][j]);
        }
        printf("\n");
    }



    for (i=0; i < row; ++i)
    {
        for (j=row-1, k=0; j>=0; --j, ++k)
        {
```

```
            mat1 [k][i] = mat[i][j];
        }
    }

    printf("\nThe 90-degree rotated matrix is: \n");
    for (i=0; i<row; ++i)
    {
        for (j=0 ; j<row; ++j)
        {
            printf("%d\t", mat1[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

OUTPUT:

```
Enter the number of rows for your matrix: 3
Enter the number of columns for your matrix: 3
Randomly Entering values in matrix... Please Wait...
Vlaues successfully stored in the matrix... This is the matrix created
13      16      25
57      53      14
45      6       37

The 90-degree rotated matrix is:
25      14      37
16      53      6
13      57      45
```

```c
// Q.2. Write a program to print a matrix in spiral order.

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


int main()
{
    int **mat, row, col, i, j, k=0, l=0;

    printf ("Enter the number of rows for your matrix: ");
    scanf ("%d", &row);
    mat=(int **) malloc (row * sizeof(int));

    printf("Enter the number of columns for your matrix: ");
    scanf ("%d", &col);
    for (i=0; i<row; ++i)
        mat [i] = (int *) malloc (col * sizeof(int));


    printf ("Randomly Entering values in matrix... Please Wait...");

    srand (time(0));
    for (i=0; i<row; ++i)
        for(j=0; j<col; ++j)
        {
            mat[i][j] = (rand() % (100));
        }
    printf("\nVlaues successfully stored in the matrix... This is the matrix created \n");
    for (i=0; i<row; ++i)
    {
        for (j=0 ; j<row; ++j)
        {
            printf("%d\t", mat[i][j]);
        }
        printf("\n");
    }

    printf ("The Spiral Matrix Output is: ");

    while (k < row && l < col)
    {
       for (i = l; i < col; ++i)
        {
            printf ("%d ", mat [k][i]);
        }
        k++;
        for (i = k; i < row; ++i)
        {
```

```
            printf ("%d ", mat[i][col - 1]);
        }
        col--;
        if (k < row)
        {
            for (i = col - 1; i >= l; --i)
            {
                printf ("%d ", mat[row - 1][i]);
            }
            row--;
        }

        if (l < col)
        {
            for (i = row - 1; i >= k; --i)
            {
                printf ("%d ", mat[i][l]);
            }
            l++;
        }
    }
    return 0;
}
```

OUTPUT:

```
Enter the number of rows for your matrix: 3
Enter the number of columns for your matrix: 3
Randomly Entering values in matrix... Please Wait...
Vlaues successfully stored in the matrix... This is the matrix created
66      28      25
95      19      5
30      89      62
The Spiral Matrix Output is: 66 28 25 5 62 89 30 95 19
```

```c
//Q.3. Given an array arr[] of positive integers of size N. Reverse every sub-array of K group elements.

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

int ul = 100;
int ll = 0;

void rvereseArray(int *arr, int start, int end)
{
    while (start < end)
    {
        int temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;
        start++;
        end--;
    }
}

int main(){
int n,k;
printf("Size of array : ");
scanf("%d", &n);
printf("Grouping No. 'K':");
scanf("%d",&k);

if(k>n){
    printf("\nK cannot be greater than N\n");
    return 0;
}

int *arr = (int*)malloc(n*sizeof(int));
int i,j;

srand(time(0));
for(i=0;i<n;i++){
    arr[i] = (rand()%(100));
}
printf("\n Generating Random array : ");
for(i=0;i<n;i++){
   printf("%d ", arr[i]);
}
printf("\n");
int c;
int* subarr = (int*)malloc(k*sizeof(int));
for(i=0;i<=n-k;i++){
    c = 0;
    for(j=i;j<i+k;j++){
```

```
        subarr[c] = arr[j];
        c++;
    }
    printf("\nthis is the subarray : ");
    for(j=0;j<k;j++){
        printf("%d ", subarr[j]);
    }
    printf("\nThe reverse is :");
    rvereseArray(subarr, 0, k-1);
    for(j=0;j<k;j++){
        printf("%d ", subarr[j]);
    }
    printf("\n");

}

return 0;
}
```

OUTPUT:

```
Size of array : 9
Grouping No. 'K':4

 Generating Random array : 48 21 85 56 77 92 72 80 45

this is the subarray : 48 21 85 56
The reverse is :56 85 21 48

this is the subarray : 21 85 56 77
The reverse is :77 56 85 21

this is the subarray : 85 56 77 92
The reverse is :92 77 56 85

this is the subarray : 56 77 92 72
The reverse is :72 92 77 56

this is the subarray : 77 92 72 80
The reverse is :80 72 92 77

this is the subarray : 92 72 80 45
The reverse is :45 80 72 92
```