ASSIGNMENT NO:3 **DATE:13/02/2015**

PROGRAM TITLE:Consider a 2D matrix of length m x n. Find all possible Saddle points from that matrix.

THEORY: An element is called the saddlepoint of the matrix if it is both the lowest element in it's row and also the highest element in it's column.

PROGRAM ALGORITHM:

```
Algo_saddlepoint(mat,m,n)
{
      for(i=0 to m)
      {
            check if lowest element in row is also the highest element in
that column;
            if(they are same)
                  print saddlepoint;
      }
      if(no saddlepoint found)
            print suitable message;
}
```

PROGRAM CODE:

```
/* C Program to find all possible Saddlepoints in a Matrix*/
#include <stdio.h>
#include <stdlib.h>
int saddlepoint(int **a,int m,int n);
int display(int **mat,int r,int c);
int main()
{
      int r,c,i,j,ele,count=0;
      printf("\n\tEnter size ::");
      scanf("%d %d",&r,&c);

      /*Allocate 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;
            }
      }

      /*Read user input*/
```

```
	printf("\n\tEnter elements::");
	for(i=0;i<r;i++)
	{
		for(j=0;j<c;j++)
		{
			scanf("%d",&mat[i][j]);
			ele=mat[0][0];
			if(ele==mat[i][j])
			{
				count ++;
			}
		}
	}
	printf("\n\tThe Matrix is::\n");
	display(mat,r,c);
	if(count==r*c)
	{
		printf("\n\tAll the elements are same. No Saddlepoint
exists.\n\tProgram Terminated\n");
		return 0;
	}
	saddlepoint(mat,r,c);
	for(i=0;i<r;i++)
	{
		free(mat[i]);
	}
	free(mat);
	return 0;
}

/*Function to find saddlepoints and print them*/
int saddlepoint(int **a,int m,int n)
{
	int i,j,k,l,c=0,max,min;
	int *ch=(int *)calloc(n,sizeof(int));
	for(i=0;i<m;i++)
	{

		/*Searching for the first column which has no saddlepoint*/
		for(j=0;j<n;j++)
		{
			if(ch[j]==0)
			{
				min=a[i][j];
				l=j;
				break;
			}
		}

		/*Finding the minimum in the row*/
		for(j=0;j<n;j++)
		{
			if((min>a[i][j])&&(ch[j]==0))
				{
					min=a[i][j];
```

```
                        l=j;
                    }
            }
            max=min;k=i;

            /*Finding the maximum in current column*/
            for(j=0;j<m;j++)
            {
                  if(max<a[j][l])
                        {
                              max=a[j][l];
                              k=j;
                        }
            }

            /*Checking if both are same*/
            if(max==min)
            {
                  c++;
                  printf("\n\tSaddle Point %d at ::%d,%d=%d\n",c,k,l,max);
                  ch[l]=1;
            }
      }
      if(c==0)
      {
            printf("\n\tNo saddlepoint found.\n");
      }
      return 0;
}

/*Function to Display Matrix*/
int display(int **mat,int r,int c)
{
      int i,j;
      for(i=0;i<r;i++)
      {
            for(j=0;j<c;j++)
            {
            printf("\t%d",mat[i][j]);
            }
            printf("\n");
      }
      return 0;
}
```

OUTPUT:

Set 1:

```
Enter size ::3 3

Enter elements::1 2 3 4 5 6 7 8 9

The Matrix is::
1     2     3
4     5     6
```

```
7    8    9

Saddle Point is at ::2,0=7
```

Set 2:

```
Enter size ::2 2

Enter elements::1 1 1 1

The Matrix is::
1    1
1    1

All the elements are same. No Saddlepoint exists.
Program Terminated
```

Set 3:

```
Enter size ::3 3

Enter elements::1 2 3 4 4 4 4 4 4

The Matrix is::
1    2    3
4    4    4
4    4    4

Saddle Point 1 at ::1,0=4

Saddle Point 2 at ::2,1=4
```

DISCUSSION:

The complexity of the Program is $O(n^2)$.The Program finds multiple saddlepoints.