**PROGRAM 1)** Given a boolean 2D matrix, find the number of islands.

CODE:

#include <stdio.h>

int limits(int n, int m, int i, int j){

if(i<0||i>=n){

return 0;

}

if(j<0||j>=m){

return 0;

}

return 1;

}

void dfs(int l[][50],int n,int m,int i,int j){

if(limits(n,m,i,j)&&l[i][j]==1){

l[i][j]=0;

dfs(l,n,m,i-1,j-1);

dfs(l,n,m,i-1,j);

dfs(l,n,m,i-1,j+1);

dfs(l,n,m,i,j-1);

dfs(l,n,m,i,j+1);

dfs(l,n,m,i+1,j-1);

dfs(l,n,m,i+1,j);

dfs(l,n,m,i+1,j+1);

}

}

int num\_of\_islands(int l[][50],int n,int m){

int count=0;

int i,j;

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

for(j=0;j<m;j++){

if(l[i][j]==1){

count++;

dfs(l,n,m,i,j);

}

}

}

if(count==0){

return 0;

}

return count;

}

int main(){

int i,j,n,m;

int l[50][50];

printf("Enter the number of rows and columns:\n");

scanf("%d %d",&n,&m);

printf("Enter the values:\n");

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

for(j=0;j<m;j++){

scanf("%d",&l[i][j]);

}

}

printf("\nThe total number of islands are: %d\n",num\_of\_islands(l,n,m));

return 0;

}

OUTPUT:

