

LABORATORIO 3

INGENIERIA EN COMPUTACION

MARIANA ESTEFANIA BARCENAS RODRIGUEZ

UAZ 3°C

ACTIVIDAD 5:

```
//mariana estefania barcenaz rodriguez
// 25022022
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>

#define m 3
#define n 3

typedef struct {
    int i;
    int j;
}Indij;

int matriz[m][n];
Indij indij;
int dato =0;

bool insertarDesordenado(int);
bool EliminaDesorden(int);
void imprimir();
Indij BuscaSec(int);
bool EsVacía();
bool ModificaDesordenado(int,int);
void OrdenarDesordenado();

main(int argc, char *argv[]){
    int opcion2;
    int opcion;
    int dato2;

    int intentar=true;
    while(intentar==true){
        do{
            printf("1...Insertar\n");
            printf("2...Eliminar\n");
            printf("3...Visualizar\n");
            printf("4...Modificar\n");
            printf("5....Ordenar\n");
            printf("0...Salir\n Elige una opcion\n");
            scanf("%c",&opcion2);
```

```

//fflush(stdin);

switch(opcion2) {
    case '1':

        printf("Dame el dato a insertar\n");
        scanf("%d",&dato);
        if(insertarDesordenado(dato)==true){
            printf("Se inserto el elemento\n");
        }else
            printf("No se inserto el elemento\n");

        break;
    case '2':
        printf("\nDame el elemento a eliminar");
        scanf("%d", &dato);
        fflush(stdin);
        if (EliminaDesorden(dato) == true) {
            printf("El elemento correctamente\n");
        } else {
            printf("El elemento no se elimino\n");
        }
        break;
    case '3':
        imprimir();
        break;
    case '4':
        printf("\nDame el elemento a buscar");
        scanf("%d", &dato2);
        fflush(stdin);
        printf("\nDame el elemento nuevo");
        scanf("%d", &dato);
        fflush(stdin);
        if (ModificaDesordenado(dato2, dato) == true) {
            printf("Se modifiko el elemento");
        } else {
            printf("El elemento no se modifiko");
        }
        break;
    case '5':
        OrdenarDesordenado();
        printf("");
        break;
}
}while(opcion2!=0);

```

```

}
}

bool EsVacia(){
    if(indij.i==0 && indij.j==0){
        if(matriz[indij.i][indij.j]==0){
            return true;
        }
        else{
            return false;
        }
    }else{
        return false;
    }
}

bool insertarDesordenado(int d ){
    if(indij.i<m){
        if(indij.j<n){
            matriz[indij.i][indij.j]=d;
            indij.j++;
            return true;
        }
        else{
            indij.j=0;
            indij.i++;
            insertarDesordenado(d);
        }

    }else{
        indij.i--;
        indij.j=n;
        return false;
    }
}

Indij BuscaSec(int elemento){
    Indij p;
    p.i = 0;
    p.j = 0;
    while(p.i <= m && p.j <= n){
        if( matriz[p.i][p.j] != elemento ){
            if( p.j == n - 1){
                p.j = 0;
                p.i++;
            }else {
                p.j++;
            }
        }
    }
}

```

```

        }
    }else {
        return p;
    }
}
return p;
}

bool EliminaDesorden(int d){
    Indij pos;
    int i,j;
    if(EsVacía()==false){
        pos=BuscaSec(d);
        if(pos.i>m||pos.j>n){
            return false;
        }else{
            i=pos.i;
            j=pos.j;
            while (i<=indij.i){
                while (j<n){
                    if(i<m && j==n-1){
                        matriz[i][j]=matriz[i+1][0];
                        i=0;
                        i++;
                    }else{
                        matriz[i][j]=matriz[i][j+1];
                        j++;
                    }
                }
                i++;
            }
            if(indij.j==0){
                indij.j=n-1;
                indij.i--;
            }else{
                indij.j--;
            }
            return true;
        }
    }else{
        return false;
    }
}
}

```

```

bool ModificaDesordenado(int datob, int dato){
    Indij pos;
    int i,j;
    if (EsVacía()==false){
        pos=BuscaSec(datob);
        if (pos.i>m || pos.j>n){
            return false;
        }else{
            i=pos.i;
            j=pos.j;
            matriz[i][j]=dato;
            return true;
        }
    }else{
        return false;
    }
}

void OrdenarDesordenado(){
    int aux;
    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            for(int k=0;k<m;k++){
                for(int l=0;l<n;l++){
                    if (matriz[i][j]<matriz[k][l]){
                        aux=matriz[i][j];
                        matriz[i][j]=matriz[k][l];
                        matriz[k][l]=aux;
                    }
                }
            }
        }
    }
}

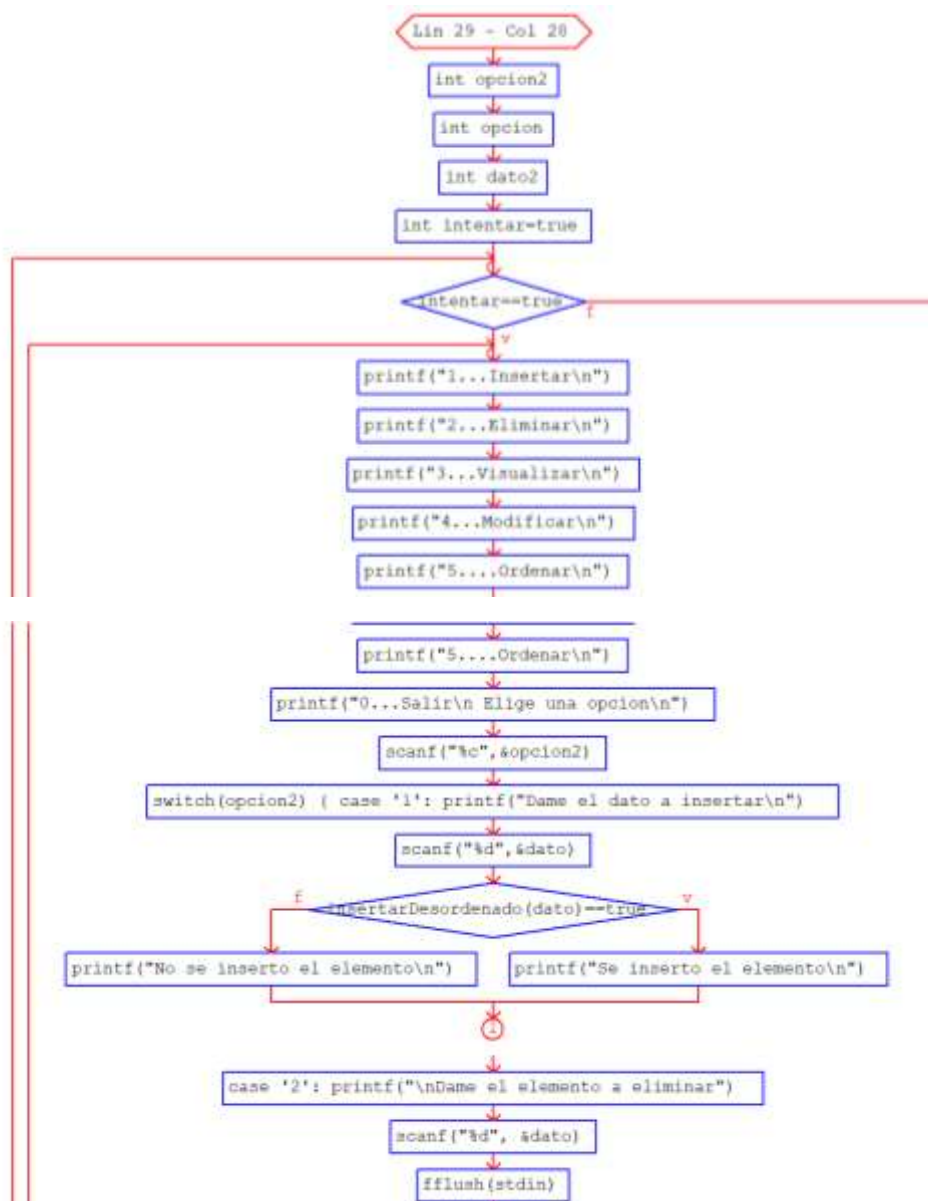
void imprimir(){
    printf("\n arreglo Bidimensional ");

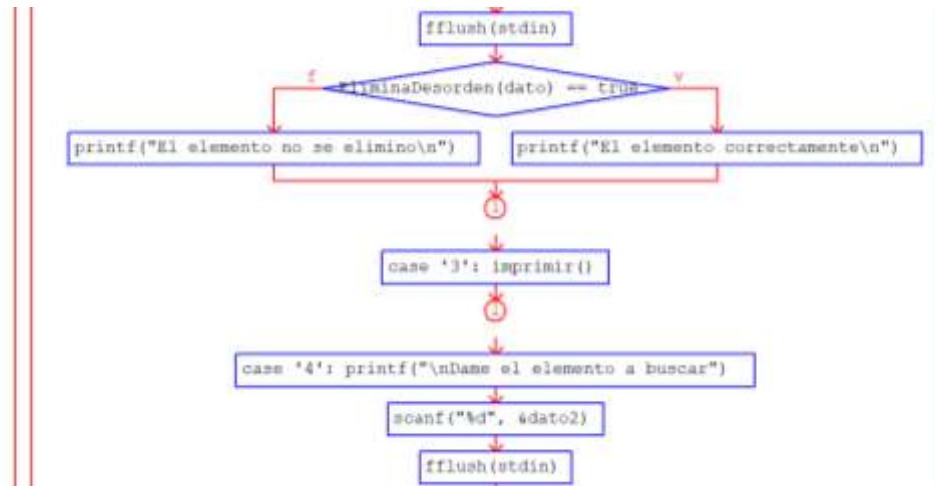
    for(int i=0;i<m;i++){
        for (int j=0;j<n;j++)
            printf("%d \t",matriz[i][j]);
        printf("\n");
    }
}

```

}

DIAGRAMA FLUJO:





ACTIVIDAD 6:

NOTA:SE SUPONE QUE UNA MATRIZ POCO DENSA ES AQUELLA EN LA CUAL HAY MAS O QUE OTRA COSA, PUES EN TEORIA CREO QUE ESO HICE MAESTRA, EN MI PROGRAMA PIDEN LOS NUMEROS QUE SE QUIEREN AGREGAR Y AL COMPILARSE HAGO QUE ADEMAS DE ESOS NUMEROS LA MATRIZ SE RELLENE DE 0.

```

//LP
// objetivo:
//mariana estefania barcenaz rodriguez
// 25/03/2022
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n,m;

    printf("\n dame el numero de filas: ");
    scanf("%d",&n);

```