

LABORATORIO 5

INGENIERIA EN COMPUTACION

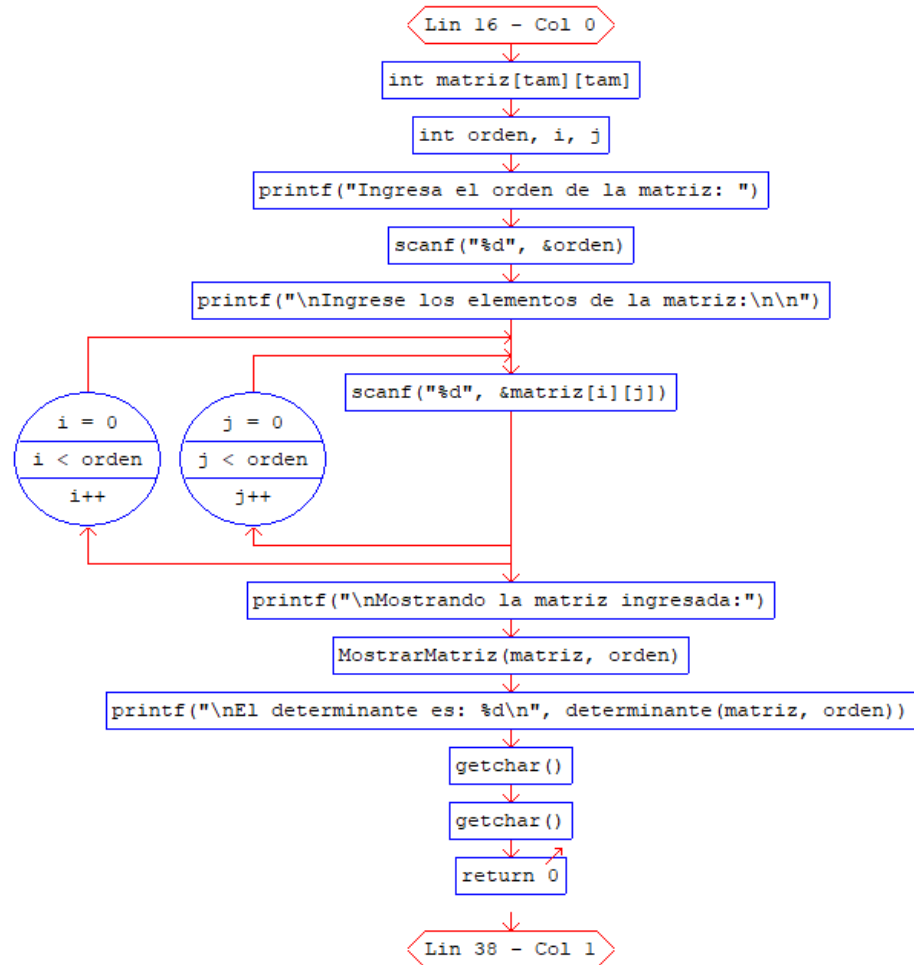
MARIANA ESTEFANIA BARCENAS RODRIGUEZ

UAZ 3°A

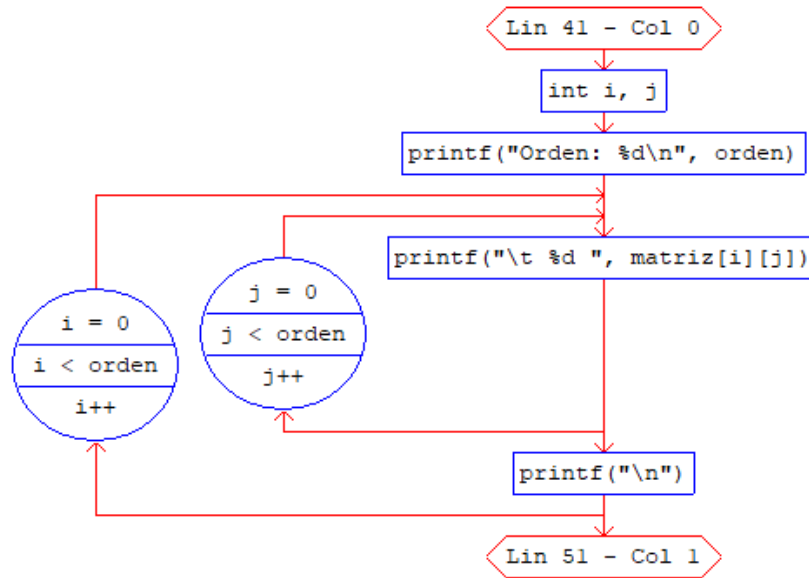
Actividad 8.- Diseñe e implemente un método recursivo que nos permita obtener el determinante de una matriz cuadrada de dimensión n.

Diagrama de Flujo:

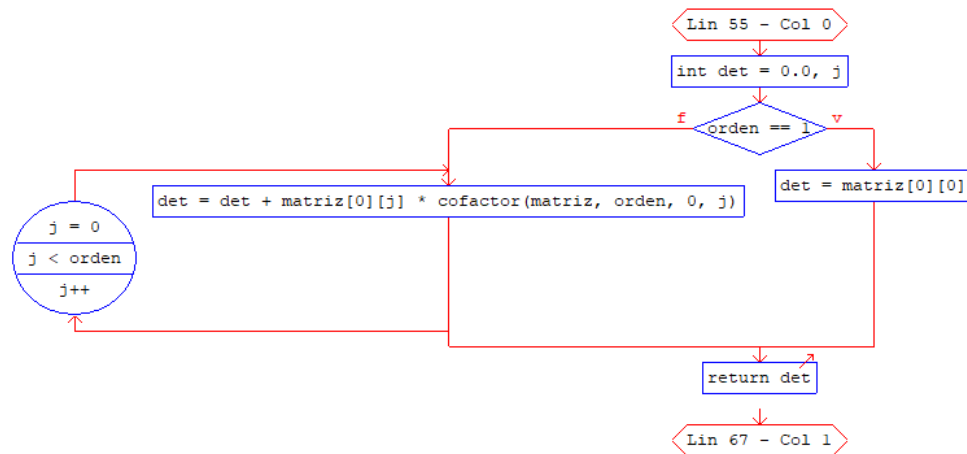
main:



MostrarMatriz



Determinante:



Cofactor:



```
/* EQUIPO:
    Victor Manuel Castillo Lujan
    Mariana Estefania Barcenás Rodríguez
*/
// FECHA: 11/03/2022

#include <stdio.h>
#include <math.h>
#define tam 20

void MostrarMatriz(int matriz[][tam], int orden);
int determinante(int matriz[][tam], int orden);
int cofactor(int matriz[][tam], int orden, int fila, int columna);
```

```

int main()
{
    int matriz[tam][tam];
    int orden, i, j;

    printf("Ingresa el orden de la matriz: ");
    scanf("%d", &orden);

    printf("\nIngresa los elementos de la matriz:\n\n");
    for (i = 0; i < orden; i++) {
        for (j = 0; j < orden; j++) {
            scanf("%d", &matriz[i][j]);
        }
    }

    printf("\nMostrando la matriz ingresada:");
    MostrarMatriz(matriz, orden);

    printf("\nEl determinante es: %d\n", determinante(matriz, orden));

    getchar();
    getchar();
    return 0;
}

void MostrarMatriz(int matriz[][tam], int orden)
{
    int i, j;

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

int determinante(int matriz[][tam], int orden)
{
    int det = 0.0, j;

    if (orden == 1) {
        det = matriz[0][0];
    }
}

```

```

    } else {
        for (j = 0; j < orden; j++) {
            det = det + matriz[0][j] * cofactor(matriz, orden, 0, j);
        }
    }

    return det;
}

int cofactor(int matriz[][tam], int orden, int fila, int columna)
{
    int submatriz[tam][tam];
    int n = orden - 1;
    int i, j;

    int x = 0;
    int y = 0;
    for (i = 0; i < orden; i++) {
        for (j = 0; j < orden; j++) {
            if (i != fila && j != columna) {
                submatriz[x][y] = matriz[i][j];
                y++;
                if (y >= n) {
                    x++;
                    y = 0;
                }
            }
        }
    }

    return pow(-1.0, fila + columna) * determinante(submatriz, n);
}

```

Actividad 9.- Dado el siguiente algoritmo recursivo:

```
void f(int num, int div) {  
    if (num>1) {  
        if ((num%div) == 0) {  
            escribir(div);  
            f(num/div,div);  
        } else {  
            f(num, div+1); } } }
```

Contestar:

- a) Dado un número cualquiera x, ¿qué nos muestra por pantalla la llamada a la función recursiva f(x,2)? ¿Cuál sería un nombre más adecuado para la función f?

NumerosPrimos

- b) Implemente un algoritmo iterativo equivalente y compare el tiempo de ejecución

```
//Mariana Estefania Barcenás Rodríguez  
// FECHA: 11/03/2022  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <conio.h>  
#include <stdbool.h>  
  
bool PrimNumRecursivo(int num, int div);  
  
int main()  
{  
  
    int n;  
  
    printf("\n Se le indicara con un 1 si es un numero primo, en cambio si  
le sale un 0 significa que no es numero primo");  
    printf("\nIngresa un numero: ");  
    scanf("%d",&n);  
  
    printf("%d", PrimNumRecursivo(n,2));  
  
}
```

```

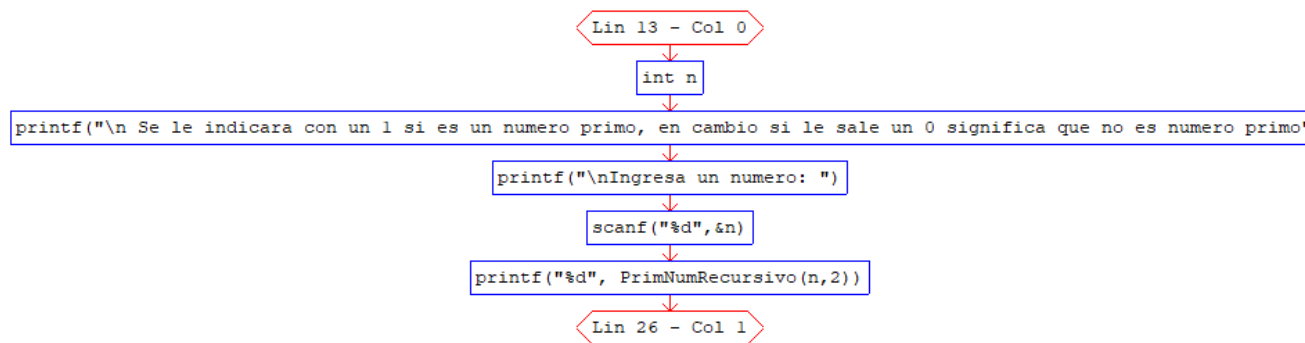
bool PrimNumRecursivo(int num, int div) {
    if (num/2 < div)
    {
        return true;
    }
    else
    {
        if ((num%div) == 0)
        {
            return false;
        }

        else
        {
            PrimNumRecursivo(num,div+1);
        }
    }
}

```

Diagrama de Flujo:

main:



PrimNumRecursivo:

