

**HERNÁNDEZ ÁVILA VÍCTOR MANUEL**

**PRACTICA 4**

**UN MENÚ CON OPERACIONES SOBRE UN ARREGLO BIDIMENSIONAL.**

**PUNTO 1**

```
#include <stdio.h>
#include <windows.h>

void funcion01();
void funcion02();
void funcion03();
void funcion04();
void funcion05();

int main()
{
    int opcion;
    do{
        printf("1. Func01\n");
        printf("2. Func02\n");
        printf("3. Func03\n");
        printf("4. Func04\n");
        printf("5. Func05\n");
        printf("6. Terminar\n");
        printf("\nProporcione la opcion deseada: ");
        scanf("%d",&opcion);
        system("cls");
        switch(opcion){
            case 1: funcion01();break;
            case 2: funcion02();break;
            case 3: funcion03();break;
            case 4: funcion04();break;
            case 5: funcion05();break;
        }
    }
    while(opcion!=6);
    return 0;
}

void funcion01(){
    printf("Funcion 01\n\n");
}

void funcion02(){
    printf("Funcion 02\n\n");
}

void funcion03(){
    printf("Funcion 03\n\n");
}

void funcion04(){
    printf("Funcion 04\n\n");
}

void funcion05(){
    printf("Funcion 05\n\n");
}
```

## Punto 2

-Se declaro el prototipo de la función Capturar que recibe como parámetros dos apuntadores (n1 y m1 respectivamente) y una matriz de 10x10 tipo flotante. La función no retorna algún valor.

-Se declaro el prototipo de la función Imprimir que recibe como parámetros dos valores enteros (n1 y m1 respectivamente) y una matriz de 10x10 tipo flotante. La función no retorna algún valor.

-En la función main se declaro una matriz llamada fMat de 10x10 tipo flotante y las variables n y m tipo enteras.

-La instrucción `system("cls");` limpia la pantalla después de seleccionar una opción.

-Dentro de la instrucción switch, en el caso 1 se llama a la función Capturar y se le pasan como parámetros la direcciones de las variables n y m, y también la matriz fMat.

-Dentro de la instrucción switch, en el caso 2 se llama a la función Imprimir y se le pasan como parámetros los valores de las variables n y m, y también la matriz fMat.

-Dentro de la definición de la función Capturar, está recibe como parámetros las direcciones de las variables n y m y las guarda en los apuntadores \*n1 y \*m1 respectivamente, también recibe la direccion de la matriz fMat y lo guarda en la matriz fMat01. Después solicita al usuario la cantidad de renglones y columnas que tendrá la Matriz que desea ingresar y guarda los valores en la variable a la cual apuntan \*n1 y \*m1, es decir las variables n y m de la función main. Estos últimos son utilizados como valores de paro en un ciclo for para capturar datos en la matriz fMat (de la función main) a través de la referencia fMat01.

-Dentro de la definición de la función Imprimir, está recibe como parámetros los valores de n,m y fMat (enviados desde la función main) y los guarda en las variables n1,m1 y fMat01 respectivamente. Utiliza como valores de paro a n1 y m1 en un ciclo for para imprimir el contenido de fMat01.

## PUNTO 3

-Se declaro el prototipo de la funcion R2\_R2MCteR1 que recibe como parámetros valores enteros que se guardaran en las variables n1,m1,R1,R2 y valores flotantes que se guardaran en las variables fCte y fMat01 respectivamente.

-Se le solicita al usuario lo renglones R1, R2 y el valor de una constante con los cuales se realizara la operación.

-Dentro de la instrucción switch, en el caso 5 se llama a la función R2\_R2MCteR1 y se le pasan como parámetros los valores de n,m, iRen1, iRen2, fCte, fMat.

-En la definición de la función R2\_R2MCteR1, recibe como parámetros los valores de n,m, iRen1, iRen2, fCte, fMat (enviados desde la función main) y los guarda en las variables n1,m1,R1,R2,fCte y fMat01 respectivamente.

-La función R2\_R2MCteR1 tiene dos variables locales i y fTerm, utiliza como valor de paro el valor de m1 en un ciclo for donde en cada iteración fTerm toma el valor de la constante fCte y multiplica al valor de la matriz fMat de la función main a traves de la referencia fMat01 en el renglón determinado por el valor de R1 e i, después el valor de fMat es modificado a través de la referencia fMat01 donde el reglón y columna es determinado por R2 e i siendo igual al valor previamente guardado más el valor de fTerm.

## PUNTO 4

```
#include <stdio.h>
#include <windows.h>
void Capturar(int *n1, int *m1, float fMat01[10][10]);
void Imprimir(int n1, int m1, float fMat01[10][10]);
void renglonPorConstante(int renglon, float constante, float matriz[10][10], int columnas);
void intercambiarRenglones(int primerRenglon, int segundoRenglon, float matriz[10][10], int columnas);
void R2_R2MCteR1(int n1, int m1, int R1, int R2, float fCte, float fMat01[10][10]);
```

```
int main()
{
    int opcion,n,m,iRen1,iRen2;
    float fMat[10][10],fCte;
    do{
        printf("=====\\n");
        printf("1. Capturar Matriz.\\n");
        printf("2. Imprimir. \\n");
        printf("3. Multiplicar renglon por constante.\\n");
        printf("4. Intercambiar renglones\\n");
        printf("5. R2_R2MCteR1\\n");
        printf("6. Terminar\\n");
        printf("\\nProporcione la opcion deseada: ");
        scanf("%d",&opcion);
        system("cls");
        switch(opcion){
            case 1: Capturar(&n,&m,fMat);break;
            case 2: Imprimir(n,m,fMat);break;
            case 3: {
                printf("Proporcione en numero de renglon: ");
                scanf("%d",&iRen1);
                printf("Proporcione el valor de la constante: ");
                scanf("%f",&fCte);
                renglonPorConstante(iRen1,fCte,fMat,m);
                break;
            }
            case 4: {
                printf("Proporcione el primer renglon: ");
                scanf("%d",&iRen1);
                printf("Proporcione el segundo renglon: ");
                scanf("%d",&iRen2);
                intercambiarRenglones(iRen1,iRen2,fMat,m);
                break;
            }
            case 5: {
                printf("Proporcione en numero del primer renglon: ");
                scanf("%d",&iRen1);
                printf("Proporcione en numero del segundo renglon: ");
                scanf("%d",&iRen2);
                printf("Proporcione el valor de la constante: ");
                scanf("%f",&fCte);
                R2_R2MCteR1(n,m,iRen1,iRen2,fCte,fMat);
                break;
            }
        }
    }
    while(opcion!=6);
    return 0;
}
```

```
void Capturar(int *n1, int *m1, float fMat01[10][10]){
    int i,j;
    printf("Proporcione la cantidad de renglones: ");
    scanf("%d",&*n1);
    printf("Proporcione la cantidad de columnas: ");
    scanf("%d",&*m1);
    for(i=0;i<*n1;i++){
        for(j=0;j<*m1;j++){
            printf("Proporcione el valor de M[%d][%d]: ",i,j);
            scanf("%f",&fMat01[i][j]);
        }
    }
}
```

```

void Imprimir(int n1, int m1, float fMat01[10][10]){
    int i,j;
    for(i=0;i<n1;i++){
        for(j=0;j<m1;j++){
            printf("%2.2ft",fMat01[i][j]);
        }
        printf("\n");
    }
}

```

```

void renglonPorConstante(int renglon, float constante, float matriz[10][10],int columnas){
    int i;
    for(i=0;i<columnas;i++){
        matriz[renglon][i]*=constante;
    }
}

```

```

void intercambiarRenglon(int primerRenglon, int segundoRenglon,float matriz[10][10],int columnas){
    int i;
    float auxiliar;
    for(i=0;i<columnas;i++){
        auxiliar=matriz[primerRenglon][i];
        matriz[primerRenglon][i]=matriz[segundoRenglon][i];
        matriz[segundoRenglon][i]=auxiliar;
    }
}

```

```

void R2_R2MCteR1(int n1, int m1, int R1, int R2, float fCte, float fMat01[10][10]){
    int i;
    float fTerm;
    for(i=0;i<m1;i++){
        fTerm=fCte*fMat01[R1][i];
        fMat01[R2][i]=fMat01[R2][i]+fTerm;
    }
    return;
}

```