



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Karina Morales Garcia

*Asignatura:* Fundamentos de programacion

*Grupo:* 20

*No. de práctica(s):* 8

*Integrante(s):* Avila Pineda Samuel David

*No. de lista o brigada:* 06

*Semestre:* 2023-1

*Fecha de entrega:* 22 de noviembre del 2022

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

### Objetivo:

El alumno elaborará programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición.

#### Actividades

- Elaborar un programa que utilice la estructura while en la solución de un problema
- Elaborar un programa que requiera el uso de la estructura do-while para resolver un problema. Hacer la comparación con el programa anterior para distinguir las diferencias de operación entre while y do-while.
- Resolver un problema dado por el profesor que utilice la estructura for en lugar de la estructura while.

### ¿Qué son las estructuras de repetición?

También son llamadas estructuras cíclicas, iterativas o de bucles. Estas nos ayudan a ejecutar un conjunto de instrucciones de manera repetida mientras que la expresión lógica a evaluar se cumpla. Existen 3 en el lenguaje C que son las siguientes: while, do-while y for.

#### **Estructura de control repetitiva while.**

Esta primero valida la expresión lógica, si es verdadera se ejecuta el bloque de instrucciones, que están delimitadas por las llaves y regresa a validar la condición nuevamente, esto lo realiza mientras la condición sea verdadera.

Su sintaxis es la siguiente:

```
while (expresión_lógica) {  
    // Bloque de código a repetir  
    // mientras que la expresión  
    // lógica sea verdadera.  
}
```

### Programa 1.

```
#include <stdio.h>  
  
int main()  
{  
    int num, cont = 0;  
    printf("\a----- Tabla de multiplicar -----\\n");  
    printf("Ingrese un número: \\n");  
    scanf("%d", &num);  
    printf("La tabla de multiplicar del %d es:\\n", num);  
    while (++cont <= 10)  
        printf("%d x %d = %d\\n", num, cont, num*cont);  
    return 0;  
}
```

El primer ejercicio o programa que hicimos fue el que se muestra anteriormente. En este ejercicio me parece que no le tome foto, debido a que se me olvidó, sin embargo, primero lo que hice fue abrir la terminal y una vez dentro abrí un archivo con el nombre pro1.c, copie el programa y posteriormente lo compile, no se mostraban errores así que ejecute.

```

Brasil19:~ fp20alu06$ vi pro1.c
Brasil19:~ fp20alu06$ gcc pro1.c -o pro1.out
Brasil19:~ fp20alu06$ ./pro1.out
----- Tabla de multiplicar -----
Ingrese un número:
4
La tabla de multiplicar del 4 es:
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
Brasil19:~ fp20alu06$

```

Esa es la única foto que tome, bueno que encontré, en este mismo ejercicio la profesora nos dejó una actividad, que es la misma que anexare más adelante en el apartado de tareas.

## Programa 2.

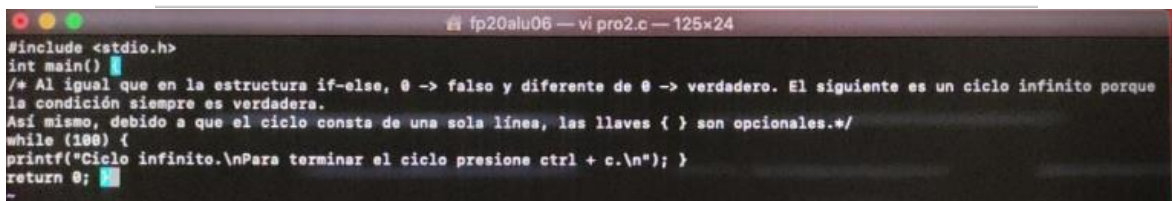
```

#include <stdio.h>
int main()
{

    /* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero.
    El siguiente es un ciclo infinito porque la condición siempre es verdadera.
    Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son
    opcionales.*/

    while (100)
    {
        printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
    }
    return 0;
}

```



```

fp20alu06 — vi pro2.c — 125x24
#include <stdio.h>
int main()
/* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero. El siguiente es un ciclo infinito porque
la condición siempre es verdadera.
Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son opcionales.*/
while (100) {
printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n"); }
return 0;

```

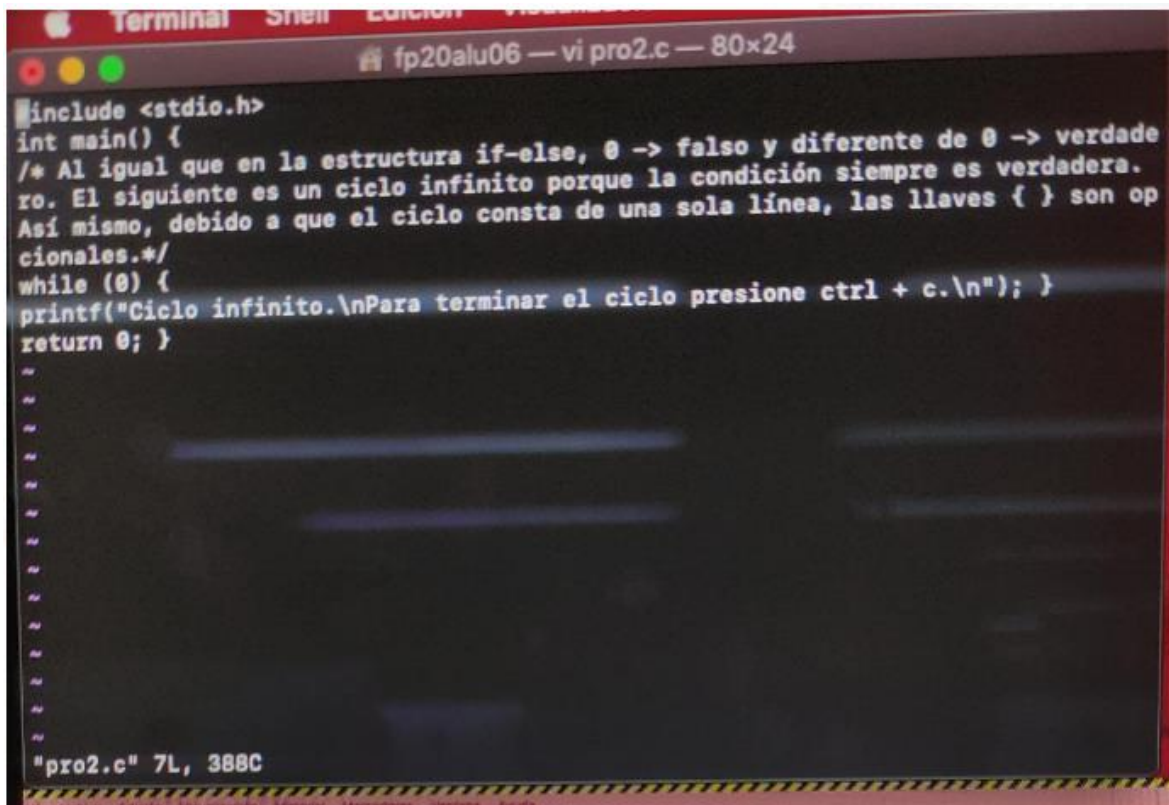
Una vez que pegue el programa en la terminal me sali del archivo para poder compilar y ejecutar.

```
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
^CPara terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
Ciclo infinito.  
Para terminar el ciclo presione ctrl + c.  
  
Brasil19:~ fp20alu06$
```

Este es el resultado a la hora de ejecutar, como podemos ver es un ciclo infinito. La profesora nos dijo que ahora teníamos que cambiar el numero 100 por cero, para ver qué pasaba.

```
Terminal Shell Edición Visualización  
fp20alu06 — vi pro2.c — 80x24  
  
#include <stdio.h>  
int main() {  
    /* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero.  
    El siguiente es un ciclo infinito porque la condición siempre es verdadera.  
    Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son opcionales.*/  
    while (0) {  
        printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n"); }  
    return 0; }
```

Una vez modificado lo anterior...

A screenshot of a terminal window with a dark background. The title bar at the top shows 'Terminal' and 'fp20alu06 — vi pro2.c — 80x24'. The code displayed is a C program that includes <stdio.h> and defines a main function. Inside main, there is a while loop with a condition of 0. The loop body contains a printf statement that says 'Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n'; and then a return 0 statement. The terminal shows the execution of the program, with the printf statement being executed repeatedly, creating a visual effect of a continuous loop. The status bar at the bottom indicates 'pro2.c' 7L, 388C.

```
#include <stdio.h>
int main() {
/* Al igual que en la estructura if-else, 0 -> falso y diferente de 0 -> verdadero. El siguiente es un ciclo infinito porque la condición siempre es verdadera. Así mismo, debido a que el ciclo consta de una sola línea, las llaves { } son opcionales.*/
while (0) {
printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n"); }
return 0; }

"pro2.c" 7L, 388C
```

Paso lo mismo que lo anterior.

### **Estructura de control repetitiva do-while.**

Es una estructura cíclica, en la que ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición, lo que quiere decir que el código se ejecuta a un determinado número de veces.

La sintaxis es:

```
do {
/*
Bloque de código que se ejecuta
por lo menos una vez y se repite
mientras la expresión lógica sea
verdadera.
*/
} while (expresión_lógica);
```

Esta estructura siempre termina con el signo de puntuación “;”

### **Programa 3.**



```

#include <stdio.h>
int main ()
{
    char op = 'n';
    double sum = 0, calif = 0;
    int veces = 0;
    do
    {
        printf("\tSuma de calificaciones\n");
        printf("Ingrese la calificación:\n");
        scanf("%lf", &calif);
        veces++;
        sum = sum + calif;
        printf("¿Desea sumar otra? S/N\n");
        setbuf(stdin, NULL); // limpia el buffer del teclado
        scanf("%c", &op);
        getchar();
    }
    while (op == 'S' || op == 's');
    printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
    return 0;
}

```

```

fp20alu06 — VI PRO3.C — 80x24
#include <stdio.h>
int main ()
{
    char op = 'n';
    double sum = 0, calif = 0; int veces = 0;
    do
    {
        printf("\tSuma de calificaciones\n");
        printf("Ingrese la calificación:\n");
        scanf("%lf", &calif);
        veces++;
        sum = sum + calif;
        printf("¿Desea sumar otra? S/N\n");
        setbuf(stdin, NULL); // limpia el buffer del teclado
        scanf("%c", &op);
        getchar();
    }
    while (op == 'S' || op == 's');
    printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
    return 0;
}
~
"PRO3.C" 22L, 470C

```

En el tercer programa lo único que hice fue copiar y pegar, pero nos marcaba errores al compilar, esto debido a que no se había pegado bien y por eso mismo se tuvo que modificar, los únicos arreglos que hice fue colocar la llave de cierre, la penúltima y agregar el signo de puntuación “;” que se encuentra ubicado en el último scanf.

```

Brasil19:~ fp20alu06$ vi pro3.c
Brasil19:~ fp20alu06$ vi pro3.c
Brasil19:~ fp20alu06$ gcc pro3.c -o pro3.out
Brasil19:~ fp20alu06$ ./pro3.out
    Suma de calificaciones
Ingrese la calificación:
7
¿Desea sumar otra? S/N
S
    Suma de calificaciones
Ingrese la calificación:
8
¿Desea sumar otra? S/N
S
    Suma de calificaciones
Ingrese la calificación:
6
¿Desea sumar otra? S/N
N
El promedio de las calificaciones ingresadas es: 7.000000
Brasil19:~ fp20alu06$

```

Eso fue lo que me arrojó como resultado y pues podemos observar que nos iba pidiendo calificaciones poco a poco y cada vez que agregábamos una calificación al final nos preguntaba si queríamos agregar otra más y nos daba dos opciones S/N que es si o no, al final nos daba el promedio de todas las calificaciones.

```

#include <stdio.h>
int main ()
{
    char op = 'n';
    double sum = 0, calif = 0; int veces = 0;
    do
    {
        printf("\tSuma de calificaciones\n");
        printf("Ingrese la calificación:\n");
        scanf("%lf", &calif);
        veces++;
        sum = sum + calif;
        printf("¿Desea sumar otra? S/N\n");
        // setbuf(stdin, NULL); // limpia el buffer del teclado
        scanf("%c",&op);
        getchar();
    }
    while (op == 'S' || op == 's');
    printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
    return 0;
}

"pro3.c" 23L, 474C

```

Una actividad extra que nos dejó la profesora con este programa es que teníamos que comentar el setbuff, como sabemos, para poder comentar lo que necesitamos es colocar dos diagonales, pero ahora lo íbamos a poner al iniciar el renglón, como se puede mostrar en la imagen.

```

Brasil19:~ fp20alu06$ vi pro3.c
Brasil19:~ fp20alu06$ gcc pro3.c -o pro3.out
Brasil19:~ fp20alu06$ ./pro3.out
Suma de calificaciones
Ingrese la calificación:
10
¿Desea sumar otra? S/N
S
El promedio de las calificaciones ingresadas es: 10.000000
Brasil19:~ fp20alu06$

```

Una vez que lo compile y ejecute, lo que me mostro fue lo mismo que anteriormente, primero me dice que ingrese una calificación, luego me pregunta si quiero agregar otra y le puse que sí, pero al momento de poner eso, ya solo me puso el promedio de todas las calificaciones.

Lo que se hizo en realidad es comentar el setbuff y al hacer esto te elimina ese paso, debido a que es solo un comentario y eso no aparece a la hora de ejecutar.

#### Programa 4.

```

#include <stdio.h>
int main ()
{
    int op, uno, dos;
    do
    {
        printf(" --- Calculadora ---\n");
        printf("\n¿Qué desea hacer\n");
        printf("1) Sumar\n");
        printf("2) Restar\n");
        printf("3) Multiplicar\n");
        printf("4) Dividir\n");
        printf("5) Salir\n");
        scanf("%d",&op);
        switch(op)
        {
            case 1:
                printf("\tSumar\n");
                printf("Introduzca los números a sumar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d + %d = %d\n", uno, dos, (uno + dos));
                break;
            case 2:
                printf("\tRestar\n");
                printf("Introduzca los números a restar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d - %d = %d\n", uno, dos, (uno - dos));
                break;
            case 3:
                printf("\tMultiplicar\n");
                printf("Introduzca los números a multiplicar separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d * %d = %d\n", uno, dos, (uno * dos));
                break;
            case 4:
                printf("\tDividir\n");
                printf("Introduzca los números a dividir separados por comas\n");
                scanf("%d, %d",&uno, &dos);
                printf("%d / %d = %.21f\n", uno, dos, ((double)uno / dos));
                break;
            case 5:
                printf("\tSalir\n");
                break;
            default:
                printf("\tOpción inválida.\n");
        }
    }
    while (op != 5);
}

```



```

fp20alu06 — vi pro4.c — 125x49
#include <stdio.h>
int main () {
int op, uno, dos; do
{
printf(" --- Calculadora ---\n");
printf("\n¿Qué desea hacer\n");
printf("1) Sumar\n");
printf("2) Restar\n");
printf("3) Multiplicar\n");
printf("4) Dividir\n");
printf("5) Salir\n");
scanf("%d",&op);
switch(op)
{
case 1:
printf("\tSumar\n");
printf("Introduzca los números a sumar separados por comas\n");
scanf("%d, %d",&uno, &dos);
printf("%d + %d = %d\n", uno, dos, (uno + dos));
break;
case 2:
printf("\tRestar\n");
printf("Introduzca los números a restar separados por comas\n");
scanf("%d, %d",&uno, &dos);
printf("%d - %d = %d\n", uno, dos, (uno - dos));
break;
case 3:
printf("\tMultiplicar\n");
printf("Introduzca los números a multiplicar separados por comas\n");
scanf("%d, %d",&uno, &dos);
printf("%d * %d = %d\n", uno, dos, (uno * dos));
break;
case 4:
printf("\tDividir\n");
printf("Introduzca los números a dividir separados por comas\n");
scanf("%d, %d",&uno, &dos);
printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos)); break;
case 5:
printf("\tSalir\n");
break;
default:
printf("\tOpcion invalida.\n");
}
}
while (op !=5);
}
-

```

Como se puede ver en la parte de arriba, copie el programa de la práctica y lo pegue en la terminal

```
Sumar
Introduzca los números a sumar separados por comas
17, 100
17 + 100 = 117
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
2

Restar
Introduzca los números a restar separados por comas
9, 5
9 - 5 = 4
--- Calculadora ---

Multiplicar
Introduzca los números a multiplicar separados por comas
11, 11
11 * 11 = 121
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
4

Dividir
Introduzca los números a dividir separados por comas
9, 4
9 / 4 = 2.25
--- Calculadora ---
```

Compile e hice unas modificaciones en el programa debido que a la hora de copiar y pegar en la terminal, lo modificaba, una vez que corregí los errores, ejecute y todo salió bien, primero me abría un menú de que era lo que quería hacer, después me decía que agregara 2 números separados por comas y dependiendo de lo que yo escogía en el menú, el programa lo realizaba.

```
brasiliv:~ #pzeal0063 ./prog4.out
--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
4
    Dividir
Introduzca los números a dividir separados por comas
9, 0
9 / 0 = inf
```

Como actividad “extra” con este programa, la profesora nos dijo que dividiéramos un numero cualquiera entre 0. A la hora de hacer eso, como resultado nos daba un número infinito, pero ¿por qué?

Esto ocurre debido a que cero es el límite de los resultados.

### Estructura de control de repetición for.

Esta nos permite realizar repeticiones cuando generalmente el control de la repetición esta definido sobre una variable contador.

La sintaxis es:

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {
    /*
        Bloque de código
        a ejecutar
    */
}
```

Esta estructura ejecuta 3 acciones básicas, dos antes y una después de ejecutar el bloque de código. La primera es la inicialización, se definen variable y se inicializan los valores. La segunda consta de la expresión lógica, se evalúa y si cumple se ejecuta el bloque, de lo contrario se continua la ejecución del programa, esto es opcional. La tercera consta de operaciones que se realizan cada vez que termina de ejecutarse el bloque de código y antes de volver a validar la expresión, igual que la anterior es opcional.

### Programa 5.

```
#include <stdio.h>
int main ()
{
    int enteroNumAlumnos = 5;
    float realCalif = 0.0, realPromedio = 0.0;
    printf("\tPromedio de calificaciones\n");
    for (int indice = 0 ; indice < enteroNumAlumnos ; indice++)
    {
        printf("\nIngrese la calificación del alumn %d\n", indice+1);
        scanf("%f",&realCalif);
        realPromedio += realCalif;
    }
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
        realPromedio/enteroNumAlumnos);
    return 0;
}
```

```

fp20alu06 — vi pro5.c — 12
#include <stdio.h>
int main ()
{
    int enteroNumAlumnos = 5;
    float realCalif = 0.0, realPromedio = 0.0;
    printf("\tPromedio de calificaciones\n");
    for (int indice = 0 ; indice < enteroNumAlumnos ; indice++)
    {
        printf("\nIngrese la calificación del alumn %d\n", indice+1);
        scanf("%f",&realCalif);
        realPromedio += realCalif;
    }
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
        realPromedio/enteroNumAlumnos);
    return 0;
}

```

En el quinto programa pues solo se copió y se pegó en la terminal, me parece que solamente se hicieron unas modificaciones en las comillas dobles y después de esto compile y ejecute.

```

Brasil19:~ fp20alu06$ vi pro5.c
Brasil19:~ fp20alu06$ gcc pro5.c -o pro5.out
Brasil19:~ fp20alu06$ ./pro5.out
    Promedio de calificaciones

Ingrese la calificación del alumn 1
8

Ingrese la calificación del alumn 2
7

Ingrese la calificación del alumn 3
8

Ingrese la calificación del alumn 4
6

Ingrese la calificación del alumn 5
10

El promedio de las calificaciones ingresadas es: 7.800000
Brasil19:~ fp20alu06$

```

Este programa como podemos ver, nos pide la calificación de 5 alumnos, uno por uno y nos imprime como resultado el promedio de dichas calificaciones.



## Define

El define nos permite definir constantes o literales. Su sintaxis es:

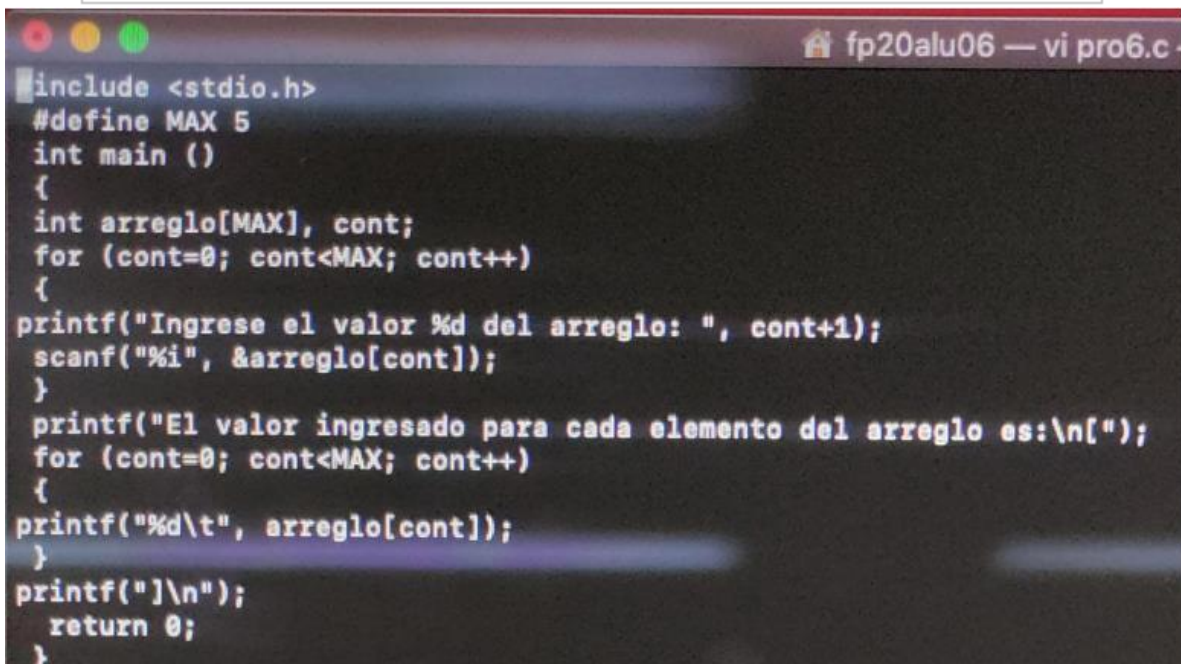
```
#define <nombre> <valor>
```

Al definir la constante simbólica #define, se emplea un nombre y un valor. Cada vez que aparezca el nombre en el programa y esto hace que se cambiara por el valor definido, puede ser numérico o por texto.

### Programa 6.

```
#include <stdio.h>
#define MAX 5
int main ()
{
    int arreglo[MAX], cont;
    for (cont=0; cont<MAX; cont++)
    {
        printf("Ingrese el valor %d del arreglo: ", cont+1);
        scanf("%i", &arreglo[cont]);
    }

    printf("El valor ingresado para cada elemento del arreglo es:\n[");
    for (cont=0; cont<MAX; cont++)
    {
        printf("%d\t", arreglo[cont]);
    }
    printf("]\n");
    return 0;
}
```



```
#include <stdio.h>
#define MAX 5
int main ()
{
    int arreglo[MAX], cont;
    for (cont=0; cont<MAX; cont++)
    {
        printf("Ingrese el valor %d del arreglo: ", cont+1);
        scanf("%i", &arreglo[cont]);
    }
    printf("El valor ingresado para cada elemento del arreglo es:\n[");
    for (cont=0; cont<MAX; cont++)
    {
        printf("%d\t", arreglo[cont]);
    }
    printf("]\n");
    return 0;
}
```

Como podemos ver en el programa, aparece #define Max 5, por lo que el programa solo nos pedirá 5 valores.

```
[Brasil19:~ fp20alu06$ vi pro6.c
[Brasil19:~ fp20alu06$ gcc pro6.c -o pro6.out
[Brasil19:~ fp20alu06$ ./pro6.out
Ingrese el valor 1 del arreglo: 6
Ingrese el valor 2 del arreglo: 7
Ingrese el valor 3 del arreglo: 2
Ingrese el valor 4 del arreglo: 5
Ingrese el valor 5 del arreglo: 6
El valor ingresado para cada elemento del arreglo es:
[6      7      2      5      6      ]
Brasil19:~ fp20alu06$
```

A la hora de compilar y ejecutar, podemos ver que en realidad si nos pidieron solo los 5 valores.

La profesora nos dijo que lo corrigiéramos, que pusiéramos otro valor en el define, no encuentro la foto, pero si a la hora de compilar y ejecutar.

```
[Brasil19:~ fp20alu06$ vi pro6.c
[Brasil19:~ fp20alu06$ gcc pro6.c -o pro6.out
[Brasil19:~ fp20alu06$ ./pro6.out
Ingrese el valor 1 del arreglo: 9
Ingrese el valor 2 del arreglo: 8
Ingrese el valor 3 del arreglo: 6
El valor ingresado para cada elemento del arreglo es:
[9      8      6      ]
Brasil19:~ fp20alu06$
```

Aquí podemos ver que solo nos define 3 valores nada más.

## Tarea

1.- Solicitar radio a usuario y calcular el área.

```
9  #include <stdio.h>
10 int main()
11 {
12     float area, radio;
13     printf("\n Introduzca el radio del circulo");
14     scanf("%f", &radio);
15     area= 3.1416 * radio * radio;
16     printf("\n El area es: %.2f", area);
17     return 0;
18 }
```

Introduzca el radio del circulo6.5

El area es: 132.73

...Program finished with exit code 0  
Press ENTER to exit console.

2.- Mostrar la numeración del 10 al 1 haciendo uso del ciclo while

```
9  #include <stdio.h>
10 int main()
11 {
12     int contador=10;
13     while(contador>0)
14     {
15         printf("%d", contador),
16         contador = contador-1 ;
17     }
18     return 0;
19 }
```

10987654321

...Program finished with exit code 0  
Press ENTER to exit console.

3.- Mostrar la tabla de multiplicar del número que ingrese el usuario con el ciclo for.

Este ejercicio se elaboró en la práctica, en la escuela, pero no le tome fotos, solo a la hora de ejecutar. Anexo la foto para que vea que si lo hice.

```
Brasil19:~ fp20alu06$ vi pro1.c
Brasil19:~ fp20alu06$ gcc pro1.c -o pro1.out
Brasil19:~ fp20alu06$ ./pro1.out
----- Tabla de multiplicar -----
Ingrese un número:
8
La tabla de multiplicar del 8 es:
8 x 0 = 0
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80
Brasil19:~ fp20alu06$
```



```

9  #include <stdio.h>
10 int main()
11 {
12     int num, cont = 0;
13     printf("\a----- Tabla de multiplicar ----- \n");
14     printf("Ingrese un número: \n");
15     scanf("%d", &num);
16     printf("La tabla de multiplicar del %d es:\n", num);
17     for(cont=1; ++cont <= 10;)
18     printf("%d x %d = %d\n", num, cont, num*cont);
19     return 0;
20 }

```

```

----- Tabla de multiplicar -----
Ingrese un número:
8
La tabla de multiplicar del 8 es:
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80

...Program finished with exit code 0
Press ENTER to exit console.

```

La foto anterior muestra que volví a copiar, pegar y modificar el programa, pero ahora desde el dispositivo de mi casa.

4.- Genera un menú con los 3 ejercicios anteriores y agrega el ciclo do-while con el fin de que el usuario no pueda salir del programa hasta seleccionar la opción en el menú

```

11 #include <stdio.h>
12
13 int main()
14 {
15     int num2, cont, num;
16     float radio, area;
17     do
18     {
19         printf;
20         printf("1) Sacar el area");
21         printf("2) Obtener la numeracion del 10 al 1");
22         printf("3) Tabla de multiplar");
23         scanf ("%d", &num2);
24     }
25     while(num==1|num==2|num==3);
26     switch(num2)
27     {
28         case 1:
29             printf("Ingresa el radio");
30             scanf("%f", &radio);
31             area=3.1416*radio*radio;
32             printf("El area es: \n", area);
33             break;
34         case 2:
35             int cont=10;
36             while(contador>0)
37                 printf("%d", contador);
38                 contador = contador-1;
39             break;
40         case 3:
41             cont=0;
42             printf("Ingresa un numero");
43             scanf("%d", &num);
44             printf("La tabla de multiplicar del %d es: \n", num);
45             for(cont=0; cont<=10; ++cont)
46                 printf("%d x %d =%d\n", num, cont, num*cont);
47             break;
48     }
49
50     return 0;
51 }

```

input

Compilation failed due to following error(s).

```

main.c: In function 'main':
main.c:32:16: warning: too many arguments for format [-Wformat-extra-args]
32 |         printf("El area es: \n", area);
    |         ^~~~~~
main.c:35:9: error: a label can only be part of a statement and a declaration is not a statement
35 |         int cont=10;
    |         ^~~~~~
main.c:36:15: error: 'contador' undeclared (first use in this function)
36 |         while(contador>0)
    |         ^~~~~~
main.c:36:15: note: each undeclared identifier is reported only once for each function it appears in

```

Solo me marco dos errores al compila ¿Cree que me pueda explicar a qué se debe esto?

5.- Del ejercicio de la calculadora, agregar condición a la división, si hay división entre cero pedir otro valor al usuario hasta que coloque un valor diferente y se pueda realizar la división.

```
9  #include <stdio.h>
10
11  int main()
12  {
13      int uno, dos;
14      do
15      {
16          printf("\tDivision\n");
17          printf("Introduzca los numeros a dividir separados por comas\n");
18          scanf("%d, %d", &uno, &dos);
19      }
20      while(uno!=0 || dos!=0);
21      printf("%d/%d=%.2lf\n", uno, dos, ((double)uno / dos));
22
23  }
```

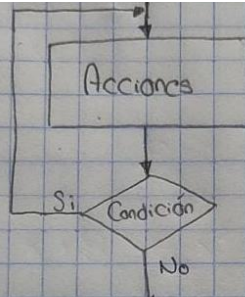
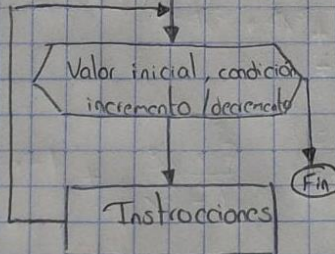
6.- Modificar el programa para que permita solicitar al usuario el Número de alumnos a evaluar empleando un ciclo while en lugar del ciclo for

```
#include <stdio.h>
int main ()
{
    int enteroNumAlumnos = 5;
    float realCalif = 0.0, realPromedio = 0.0;
    printf("\tPromedio de calificaciones\n");
    for (int indice = 0 ; indice < enteroNumAlumnos ; indice++)
    {
        printf("\nIngrese la calificación del alumn %d\n", indice+1);
        scanf("%f",&realCalif);
        realPromedio += realCalif;
    }
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
        realPromedio/enteroNumAlumnos);
    return 0;
}
```

```
9  #include <stdio.h>
10
11  int main()
12  {
13      int enteroNumAlumnos=5, indice=0;
14      float realCalif=0.0, realPromedio=0.0;
15      printf("\tPromedio de calificaciones\n");
16      while(indice<enteroNumAlumnos)
17      {
18          printf("\nIngresa la calificacion del alumno%d\n", indice+1);
19          scanf("%f", &realCalif);
20      }
21      printf("\nEl promedio de las calificaciones ingresadas es: %f\n", realPromedio/enteroNumAlumnos);
22  }
```

## 7.- Completa el cuadro con las estructuras iterativas (MIENTRAS, HACER MIENTRAS Y PARA)

Estructura	Diagrama	Pseudocódigo	Codificación
Iterativa Mientras		<b>MIENTRAS</b> que <expresión lógica> Instrucciones <b>FIN DEL MIENTRAS</b>	While(expresión) { Secuencia de instrucciones; }

Iterativa de Hacer Mientras		Hacer Instrucciones Mientras- que <expresión lógica> Fin del Hacer Mientras	do { Secuencia; } while (expresión)
Iterativa Para		Para (valor inicial, condición, incremento/decremento) Instrucciones Fin Para	for (valor inicial, condición, incremento/decremento) { Secuencia; }

### Conclusiones

Esta práctica, se me complicó un poco al pasar de while a for o viceversa, aun no me queda del todo claro cómo se hace.

Gracias a esta práctica puedo concluir que es de vital importancia saber cómo es que funcionan las estructuras iterativas, esto para que poder realizar el programa, dichas estructuras nos ayudan a realizar cuantos ciclos uno quiera y poder repetir instrucciones.

### Referencias

- [https://www.onlinegdb.com/online\\_c\\_compiler](https://www.onlinegdb.com/online_c_compiler)
- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.