

	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorio de docencia	

# Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de programación

Grupo:

No. de Práctica:

Integrantes: Ever Ivan Rosales Gómez

No de Lista o brigade:

Semestre: 2023-2

Fecha de entrega: 03/05/23

Observaciones:

Calificación

## ESTRUCTURA DE REPETICIÓN

### OBJETIVO:

Elaborar programas en lenguaje C que incluyan estructuras de repetición para la resolución de problemas.

### DESARROLLO:

Una estructura de repetición sirve para que un programa se repita varias veces mientras la condición lo cumpla.

Es importante mencionar que en este tipo de estructuras es común identificar que las expresiones tengan incremento o decremento con ayuda de ++, --, respectivamente

Existen 3 tipos de estructuras de repetición:

#### -WHILE

Se tiene una expresión lógica, la expresión WHILE primero válida si es verdadera, en caso de que se cumpla el programa se lleva a cabo, posteriormente si la expresión lógica se vuelve a cumplir y ejecutar, así hasta que se deje de cumplir.

La forma es la siguiente:

```
while (expresión lógica){

    //operaciones que va a realizar

}
```

### 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) //estructura WHILE
printf("%d x %d = %d\\n", num, cont, num*cont); return 0;
}
```

```

Last login: Wed Apr 26 19:15:03 on ttys000
Brasil19:~ fp24alu35$ vi prog1.c
Brasil19:~ fp24alu35$ gcc prog1.c -o prog1.out
Brasil19:~ fp24alu35$ ./prog1.out
----- Tabla de multiplicar -----
Ingrese un número:
10
La tabla de multiplicar del 10 es:
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
Brasil19:~ fp24alu35$ █

```

Como se puede apreciar en la foto la expresión while tendrá incremento debido a que tiene ++. Al compilar el ciclo while ayudará a obtener los 10 primeros valores de una tabla de multiplicar dado un número que arrojará el usuario.

Pero... ¿Qué pasa si la expresión no tiene incremento o decremento?

```

#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) //estructura WHILE
    printf("%d x %d = %d\\n", num, cont, num*cont); return 0;
}
█

```



Este ejercicio primero cuenta con un IF (condición simple) posteriormente se agrega la estructura repetitiva WHILE para que el valor lo multiplique 6 veces, además lleva un contador que permite sumar el resultado de esas 6 multiplicaciones. Es decir, el usuario ingresa un valor mayor a 5 y menor a 10, en caso contrario el programa no se llevará a cabo, el valor dado por el usuario se multiplica por 10, el nuevo resultado de nuevo se multiplica por 10 así por 6 veces, por lo que se tendrán 6 resultados, estos se van a sumar y son los que arroja el programa de tal manera que los va a sumar e imprimir.

### 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;
}
```

```
Brasil19:~ fp24alu35$ gcc prog3.c -o prog3.out
Brasil19:~ fp24alu35$ ./prog3.out
Suma de calificaciones
Ingrese la calificación:
9
¿Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:
9
¿Desea sumar otra? S/N
s
Suma de calificaciones
Ingrese la calificación:
9
¿Desea sumar otra? S/N
n
El promedio de las calificaciones ingresadas es: 9.000000
Brasil19:~ fp24alu35$
```

Este programa sirve para sumar una o más calificaciones y además obtener el promedio (sumar de n valores y dividirlos entre n valores). La característica es que el usuario puede elegir si quiere agregar calificaciones o no con ayuda de **s** (agregar otra calificación) o **n** (no agregar).

El while entra para conocer si se agrega o no calificación.

### -DO-WHILE

Esta estructura repetitiva puede ser un poco complicada pero basta con entender la teoría y posteriormente realizar ejercicios. Esta estructura en palabras simples **primero** realiza o ejecuta el bloque y **posteriormente** revisa la condición lógica. La sintaxis es la siguiente:

```
do {
    //INSTRUCCIONES Y SE REPITE N VECES
} while (expresión_lógica);
```

Viendo la sintaxis es importante ver el orden de las llaves, pues primero se pone el do seguido de una llave y cierra hasta el momento de poner el while con su respectiva condición.

Otra cosa importante a mencionar es que el **while** en este caso lleva punto y coma al finalizar el paréntesis.

#### EJERCICIO 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");
                } }
        while (op != 5); }
```

/\*Solicitar un valor al usuario, mayor igual a 5 y menor igual a 10  
multiplicar x 10 seis veces e imprimir la sumatoria\*/

```
#include <stdio.h>
int valor;
int i=1;
int sumatoria=0;
int main (){
    do{
        printf("\tIngresa un valor: ");
        scanf("%d",&valor);
        }while(valor>5&&valor<10);
        while(i<=6)
        {
            valor=valor*10;
            i++;
            sumatoria=sumatoria+valor;
        }
        printf("La sumatoria es = %d",sumatoria);

    printf("\n");
    return 0;
}
```

Este programa es una calculadora que permite hacer las 4 operaciones básicas con ayuda de un **menú**. El usuario puede elegir que opción quiere con ayuda de un

**switch.** Si elige 1 será Suma, 2 Resta, 3 Multiplicación, 4 División, 5 Salir. En caso de que el usuario ingrese un valor diferente a 5 (default) El programa no se ejecuta

### INDICAR LAS DIFERENCIAS ENTRE WHILE Y DO WHILE

WHILE	DO WHILE
<ul style="list-style-type: none"> <li>-No lleva punto y coma</li> <li>-Entre las llaves va el bloque de instrucción</li> <li>-Primero se ve la condición y después ejecuta</li> </ul>	<ul style="list-style-type: none"> <li>-Primero ejecuta el bloque y después ve la condición</li> <li>-Las llaves van al final del do y al principio del while</li> <li>-El while al final del paréntesis va punto y coma</li> </ul>

### -ESTRUCTURA FOR

Es el equivalente a la estructura WHILE, solo que este se usa cuando la repetición está definida.

Una ventaja de usar **for** es que se escribe en una sola línea y que se usa una variable, por ejemplo la variable se llamará i (qué es lo más común). Con la sintaxis siguiente:

Inicializar la variable		Darle un límite		Escribir si va en incremento o decremento	
i=1	;	i<=10	;	i++	;
<pre> for (i=1;i&lt;=10;i++) {     //bloque de instrucción } </pre>					

Como se puede ver en la tabla, se está usando un ciclo for en donde mi variable i es entera, le estoy asignando que inicializará desde 1 y tiene que ser igual o menor a 10 para que el bloque conozca esos extremos, por último estoy incrementando mi variable que vaya de 1 en 1.

## EJERCICIO 5.

```
#include <stdio.h>
int i=0;
int valor;
int sumatoria=0;

int main (){
    do{
        printf("\tIngresa un valor");
        scanf("%d",&valor);
    }while(valor>=5&&valor<=10);
    for(i=1;i<=6;i++) //AQUÍ VA LA ESTRUCTURA FOR
    {
        valor=valor*10;
        sumatoria=sumatoria+valor;
    }
    printf("\nLa sumatoria es = %d",sumatoria);
    return 0;
}
```

```
Brasil19:~ fp24alu35$ gcc prog6.c -o prog6.out
```

```
Brasil19:~ fp24alu35$ ./prog6.out
```

```
Ingresa un valor3
```

```
La sumatoria es = 3333330Brasil19:~ fp24alu35$
```

Se utilizó uno de los ejercicios anteriores, en donde el ciclo for está presente, la variable va desde 1 hasta que sea igual o menor a 6 y que vaya en incremento. Lo que hará el programa es pedirle al usuario un valor y ese se va a multiplicar por 10 veces obteniendo un resultado, ese resultado de nuevo se va a multiplicar por 10 hasta conseguir 6 resultados, Finalmente el programa los sumará y los mostrará en pantalla.

### -DEFINE

El define se puede considerar una estructura repetitiva pero esta se usa al declarar variables, un ejemplo para poder entender esto es cuando ocupamos printf y scanf, para imprimir en pantalla y leer la variable respectivamente. Pues para no usar en el código siempre printf y scanf se pueden sustituir por p (printf) y s (scanf), PERO para que se lleve acabo y no genere error el programa se tiene que definir, en este caso se usa **define** de la siguiente manera.

DEFINE	NOMBRE	VALOR
#define	p	printf
#define p printf		

Ahora el programa cada vez que se escriba la letra p sabrá que se habla de printf debido a que ya se definió.



## EJERCICIOS DE TAREA.

### 1. Solicitar el radio al usuario y calcular área

```
#include <stdio.h>
#include <math.h>
float radio,resultado;
float pi=3.1416;

int main (){
    printf("\tIngresa el valor del radio: ");
    scanf("%f",&radio);
    resultado=pi*pow(radio,2);
    printf("\nEl area del circulo es %f",resultado);
    printf("\n");
    return 0;
}
```

```
[Aldair:~ aldair$ gcc tarea1.c -o tarea1.out
[Aldair:~ aldair$ ./tarea1.out
    Ingresa el valor del radio: 5.25

El area del circulo es 86.590347
```

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

```
C tarea2.c > main()
1  #include <stdio.h>
2  int contador=11; //el programa pide iniciar desde 10
3  int main (){}
4      while (contador>=1){ //la condicion while menciona hasta donde va a llegar el contador
5          contador --; //Se pone -- ya que irá en DECREMENTO
6          printf("Valor es = %d\n",contador);
7      }
8  return 0;
9  }
```

```
● Aldair:aprendiendo en C aldair$ gcc tarea2.c -o tarea2.out
● Aldair:aprendiendo en C aldair$ ./tarea2.out
Valor es = 10
Valor es = 9
Valor es = 8
Valor es = 7
Valor es = 6
Valor es = 5
Valor es = 4
Valor es = 3
Valor es = 2
Valor es = 1
Valor es = 0
```

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

```

C tarea2.c > main()
1  #include <stdio.h>
2  int valor,resultado=1;
3  int i=1;
4  int main () {
5      printf("\tIngresa un valor: ");
6      scanf("%d",&valor);
7      for(i=1;i<=10;i++){
8          printf("%d x %d = %d \n",valor,i,valor*i);
9      }
10     printf("\n");
11     return 0;
12 }

```

```

● Aldair:aprendiendo en C aldair$ gcc tarea2.c -o tarea2.out
● Aldair:aprendiendo en C aldair$ ./tarea2.out
    Ingresa un valor: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

```

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

```

C tarea2.c > main()
1  #include <stdio.h>
2  #include <math.h>
3  #include <locale.h>
4  //declaracion de variables para del primer ejercicio
5  float radio, resultado_1;
6  float pi=3.1416;
7  //declaracion de variables del segundo ejercicio
8  int contador=1;
9  int i=0;
10 //declaracion de variables del tercer ejercicio
11 int valor,resultado_3=1;
12 int i_1=0;
13
14 //declaracion de variables del nuevo ejercicio
15 int opcion;
16 int main () {
17     do {
18         setlocale(LC_ALL,"ESPAÑOL");
19         printf("\t\tMENÚ");
20         printf("\n1)Obtener el radio");
21         printf("\n2)Números en decremento");
22         printf("\n3)Tabla de multiplicar");
23         printf("\n4)Salir");
24         scanf("%d",&opcion);
25         printf("\nSelecciona una opción: ");
26         switch (opcion){
27             case 1:
28                 printf("\nIngresa el valor del radio: ");
29                 scanf("%f",&radio);
30                 resultado_1=(pi)*(pow(radio,2));
31                 printf("\nEl área del círculo es %f",resultado_1);
32                 break;
33
34             case 2:
35                 for (i=1;i<=10;i++){
36                     printf("\nValor es = %d",contador);
37                     break;
38                 }
39
40             case 3:
41                 printf("\nIngresa un valor: ");

```

```

38
39                 printf("\nIngresa un valor: ");
40                 scanf("%d",&valor);
41                 for (i_1=0;i_1<=10;i_1++){
42                     printf("\n%d x %d = %d",valor,i_1,resultado_3);
43                     break;
44                 }
45             default:
46                 printf("\nVuelve a seleccionar una opción válida\n");
47                 break;
48         }
49         printf("\n");
50     }while (opcion!=3);
51     return 0;
52 }

```

```

Aldair:aprendiendo en C aldair$ gcc tarea2.c -o tarea2.out
Aldair:aprendiendo en C aldair$ ./tarea2.out
    MENÚ
1)Obtener el radio
2)Números en decremento
3)Tabla de multiplicar
4)Salir4

Selecciona una opción:
Vuelve a seleccionar una opción válida

    MENÚ
1)Obtener el radio
2)Números en decremento
3)Tabla de multiplicar
4)Salir1

Selecciona una opción:
Ingresa el valor del radio: 5.55

El área del círculo es 96.769135
    MENÚ
1)Obtener el radio
2)Números en decremento
3)Tabla de multiplicar
4)Salir[]

```

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

```

C tarea5.c > main()
1  #include <stdio.h>
2  int main ()
3  {
4      int op, uno, dos;
5      do
6      {
7          printf(" --- Calculadora ---\n");
8          printf("\n¿Qué desea hacer?\n");
9          printf("1) Sumar\n");
10         printf("2) Restar\n");
11         printf("3) Multiplicar\n");
12         printf("4) Dividir\n");
13         printf("5) Salir\n");
14         scanf("%d",&op);
15         switch(op){
16             case 1: //OPERACIONES DE SUMA
17                 printf("\tSumar\n");
18                 printf("Introduzca los números a sumar separados por comas\n");
19                 scanf("%d",&uno,&dos);
20                 printf("%d + %d = %d\n", uno, dos, (uno + dos));
21                 break;
22             case 2: //OPERACIONES DE RESTA
23                 printf("\tRestar\n");
24                 printf("Introduzca los números a restar separados por comas\n");
25                 scanf("%d",&uno,&dos);
26                 printf("%d - %d = %d\n", uno, dos, (uno - dos));
27                 break;
28             case 3: //OPERACIONES DE MULTIPLICACIÓN
29                 printf("\tMultiplicar\n");
30                 printf("Introduzca los números a multiplicar separados por comas\n");
31                 scanf("%d",&uno,&dos);
32                 printf("%d * %d = %d\n", uno, dos, (uno * dos));
33                 break;
34             case 4: //OPERACIONES DE DIVISION
35                 printf("\tDividir\n");
36                 printf("Introduzca los números a dividir separados por comas\n");
37                 scanf("%d",&uno,&dos);
38                 if (uno == 0 || dos==0) //-----ESTA ES LA CONDICIÓN QUE HACE NO HACER LA DIVISIÓN PORQUE YA SEA QUE UNO DE LOS VALORES ES 0
39                     printf("Ingresa valores diferentes a 0");
40                 else {
41                     printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
42                 }
43                 break;
44             case 5:
45                 printf("\tSalir\n");
46                 break;
47             default:
48                 printf("\tOpción inválida.\n");
49         }
50     }
51     while (op != 5);
52 }

```

```

Aldair:aprendiendo en C aldair$ gcc tarea5.c -o tarea5.out
Aldair:aprendiendo en C aldair$ ./tarea5.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
0,0
Ingresa valores diferentes a 0 Salir

```

```

--- Calculadora ---

¿Qué desea hacer
1) Sumar
2) Restar
3) Multiplicar
4) Dividir
5) Salir
4
    Dividir
Introduzca los números a dividir separados por comas
5,1
5 / 1 = 5.00
    Salir
Calculadora

```

**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**

#### Programa5.c

```

#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;
}

```

```

C tarea6.c > main()
1  #include <stdio.h>
2  int enteroAlumnos = 5;
3  int indice=0;
4  float realCalif = 0.0, realPromedio = 0.0;
5
6  int main () {
7      printf("\tPromedio de calificaciones");
8      while (indice < enteroAlumnos) { //---AQUI VA EL CICLO WHILE CON LA CONDICION QUE EL INDICE DEBE SER MENOR AL NUMERO DE ALUMNOS QUE ES 5
9          printf("\n\tIngrese la califacion del alumno %d :",indice+1);
10         scanf("%f",&realCalif);
11         indice ++; //---AQUI IRÁ EN INCREMENTO
12         realPromedio += realCalif; // EL realPromedio es igual al realPromedio MAS realCalif
13     }
14     printf("\nEl promedio de las calificaciones ingresadas es: %f",realPromedio/enteroAlumnos); //---DENTRO DEL PRINTF VA LA DIVISION
15
16     return 0;
17 }

```

```

Aldair:aprendiendo en C aldairst gcc tarea6.c -o tarea6.out
Aldair:aprendiendo en C aldairst ./tarea6.out
    Promedio de calificaciones
    Ingrese la califacion del alumno 1 :9
    Ingrese la califacion del alumno 2 :9
    Ingrese la califacion del alumno 3 :9
    Ingrese la califacion del alumno 4 :9
    Ingrese la califacion del alumno 5 :9
    El promedio de las calificaciones ingresadas es: 9.000000Aldair:aprendiendo en C aldairst

```

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

ESTRUCTURA	DIAGRAMA	PSEUDOCÓDIGO	CODIFICACIÓN
<b>MIENTRAS</b>		<b>MIENTRAS</b> < EXPRESIÓN LÓGICA >  <b>INSTRUCCIONES</b>  <b>FIN DEL</b> <b>MIENTRAS</b>	<b>WHILE</b> { (EXPRESIÓN)  <b>//INSTRUCCIONES</b> }  
<b>HACER MIENTRAS</b>		<b>HACER</b> <  <b>INSTRUCCIONES</b>  > <b>MIENTRAS</b> < EXPRESIÓN LÓGICA>  <b>FIN DEL HACER</b> <b>MIENTRAS</b>	<b>DO</b> {  <b>//INSTRUCCIONES</b>  <b>}WHILE</b> (EXPRESIÓN);  
<b>PARA</b>		<b>PARA</b> < EXPRESIÓN LÓGICA INCREMENTO/DEC REMENTO>  <b>FIN PARA</b>	<b>FOR</b> { (INICIA,LÍMITE, INCREMENTO/DEC REMENTO)  }  

## CONCLUSIONES

Es importante conocer el momento correcto para usar los 3 tipos de estructuras de repetición.

Las 3 estructuras tienen algo en común y es que se usan cuando se quiere repetir un conjunto de sentencias ya sea por un número determinado o mientras se cumpla algo. Ahora es importante saber cuándo usar una y cuándo usar otra.

Cuando sabemos el número de repeticiones que va a tener el programa se usa while. Cuando no sabemos cuantas veces se va a ejecutar se usa do while.

Finalmente, se usa el for similar a while, pero de una manera más compacta, en una sola línea, conociendo desde donde partirá la variable hasta donde va a llegar y si irá en incremento o decremento.

**REFERENCIAS**

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.