



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación

## Salas A y B

*Profesor:* Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No. De Práctica(s):* 9

*Integrante(s):* Emiliano Guevara Chávez

*No. de Equipo de  
Cómputo empleado:* No aplica

*No. De Lista o Brigada:* 20

*Semestre:* Ciclo 2021-1

*Fecha de entrega:* 15/12/20

*Observaciones:* \_\_\_\_\_

**CALIFICACIÓN:** \_\_\_\_\_

## Práctica #9: Estructuras de repetición

### Objetivo:

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva define.

### Introducción:

Las estructuras de repetición son las llamadas estructuras cíclicas, iterativas o de bucles. Permiten ejecutar un conjunto de instrucciones de manera repetida (o cíclica) mientras que la expresión lógica a evaluar se cumpla (sea verdadera). En lenguaje C existen tres estructuras de repetición: while, do-while y for. Las estructuras while y do-while son estructuras repetitivas de propósito general.

### Desarrollo de las actividades:

#### Estructura de control repetitiva while

#### Código (estructura de repetición while)

```
corewhile.c
1  #include <stdio.h>
2
3  /*
4   * Este programa genera la tabla de multiplicar de un número dado.
5   * El número se lee desde la entrada estándar (teclado).
6   */
7  int main()
8  {
9      char au=163;
10     int num, cont = 0;
11
12     printf("\a----- Tabla de multiplicar -----\\n");
13     printf("Ingrese un número: \\n",au);
14     scanf("%d", &num);
15
16     printf("La tabla de multiplicar del %d es:\\n", num);
17     while (++cont <= 10)
18     {
19         printf("%d x %d = %d\\n", num, cont, num*cont);
20     }
21     return 0;
22 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc corewhile.c -o corewhile.exe
C:\Users\user\Desktop\Lenguaje C\Ejemplos>corewhile.exe
----- Tabla de multiplicar -----
Ingrese un número:
5
La tabla de multiplicar del 5 es:
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
C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```

## Código (estructura de repetición while)

```
corewhile.c  codesrewhile.c
1  #include <stdio.h>
2  /*
3   * Este programa genera un ciclo infinito.
4   */
5  int main()
6  {
7      // Al igual que en la estructura if-else
8      // 0 -> falso
9      // diferente de 0 -> verdadero
10
11     // El siguiente es un ciclo infinito
12     // porque la condición siempre es verdadera.
13     // Así mismo, debido a que el ciclo consta de una sola línea, las
14     // llaves { } son opcionales.
15
16     while (100)
17     {
18         printf("Ciclo infinito.\nPara terminar el ciclo presione ctrl + c.\n");
19     }
20     return 0;
21 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc codesrewhile.c -o codesrewhile.exe
C:\Users\user\Desktop\Lenguaje C\Ejemplos>codesrewhile.exe
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.
```

## Estructura de control repetitiva do-while

### Código (estructura de repetición do-while)

```
repetidowhile.c
1  #include <stdio.h>
2  /*
3   * Este programa obtiene el promedio de calificaciones ingresadas por
4   * el usuario. Las calificaciones se leen desde la entrada estándar (teclado).
5   * La inserción de calificaciones termina cuando el usuario presiona una tecla
6   * diferente de 'S' o 's'.
7   */
8  int main ()
9  {
10     char ac=162;
11     char iter=168;
12     char op = 'n';
13     double sum = 0, calif = 0;
14     int veces = 0;
15     do
16     {
17         printf("\tSuma de calificaciones\n");
18         printf("Ingrese la calificación:\n",ac);
19         scanf("%lf", &calif);
20         veces++;
21         sum = sum + calif;
22
23         printf("%cDesea sumar otra? S/N\n",iter);
24         setbuf(stdin, NULL); // limpia el buffer del teclado
25         scanf("%c",&op);
26         getchar();
27     }
28     while (op == 'S' || op == 's');
29
30     printf("El promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
31     return 0;
32 }
```

```

C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc repetidowhile.c -o repetidowhile.exe
C:\Users\user\Desktop\Lenguaje C\Ejemplos>repetidowhile.exe
Suma de calificaciones
Ingrese la calificación:
8
¿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:
7
¿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: 7.500000
C:\Users\user\Desktop\Lenguaje C\Ejemplos>_

```

## Código (estructura de repetición do-while)

```

repetidowhile.c esredowhile.c
1  #include <stdio.h>
2  /* Este programa genera una calculadora básica. */
3  int main ()
4  {
5      char iter="168";
6      char au="163";
7      char ao="162";
8      char aa="160";
9      char ae="130";
10     int op, uno, dos;
11     do
12     {
13         printf(" --- Calculadora ---\n");
14         printf("\n¿Qué desea hacer\n",iter,ae);
15         printf("1) Sumar\n");
16         printf("2) Restar\n");
17         printf("3) Multiplicar\n");
18         printf("4) Dividir\n");
19         printf("5) Salir\n");
20         scanf("\n%d",&op);
21
22         switch(op)
23         {
24             case 1:
25                 printf("\tSumar\n");
26                 printf("Introduzca los números a sumar separados por comas\n",au);
27                 scanf("%d, %d",&uno, &dos);
28                 printf("%d + %d = %d\n", uno, dos, (uno + dos));
29                 break;
30             case 2:
31                 printf("\tRestar\n");
32                 printf("Introduzca los números a restar separados por comas\n",au);
33                 scanf("%d, %d",&uno, &dos);
34                 printf("%d - %d = %d\n", uno, dos, (uno - dos));
35                 break;
36             case 3:
37                 printf("\tMultiplicar\n");
38                 printf("Introduzca los números a multiplicar separados por comas\n",au);
39                 scanf("%d, %d",&uno, &dos);
40                 printf("%d * %d = %d\n", uno, dos, (uno * dos));
41                 break;
42             case 4:
43                 printf("\tDividir\n");
44                 printf("Introduzca los números a dividir separados por comas\n",au);
45                 scanf("%d, %d",&uno, &dos);
46                 printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
47                 break;
48             case 5:
49                 printf("\tSalir\n");
50                 break;
51             default:
52                 printf("\tOpción inválida.\n",ao,aa);
53         }
54     }
55     while (op != 5);
56     return 0;
57 }

```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc esredowhile.c -o esredowhile.exe
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>esredowhile.exe
```

```
--- Calculadora ---
```

```
{Qué desea hacer
```

- 1) Sumar
- 2) Restar
- 3) Multiplicar
- 4) Dividir
- 5) Salir

```
1
```

```
Sumar
```

```
Introduzca los números a sumar separados por comas
```

```
1,5
```

```
1 + 5 = 6
```

```
--- Calculadora ---
```

```
{Qué desea hacer
```

- 1) Sumar
- 2) Restar
- 3) Multiplicar
- 4) Dividir
- 5) Salir

```
2
```

```
Restar
```

```
Introduzca los números a restar separados por comas
```

```
45,65
```

```
45 - 65 = -20
```

```
--- Calculadora ---
```

```
{Qué desea hacer
```

- 1) Sumar
- 2) Restar
- 3) Multiplicar
- 4) Dividir
- 5) Salir

```
3
```

```
Multiplicar
```

```
Introduzca los números a multiplicar separados por comas
```

```
89,0
```

```
89 * 0 = 0
```

```
--- Calculadora ---
```

```
{Qué desea hacer
```

- 1) Sumar
- 2) Restar
- 3) Multiplicar
- 4) Dividir
- 5) Salir

```
4
```

```
Dividir
```

```
Introduzca los números a dividir separados por comas
```

```
34,54
```

```
34 / 54 = 0.63
```

## Estructura de control de repetición for

### Código (estructura de repetición for)

```
1  #include <stdio.h>
2  /* Este programa genera un arreglo unidimensional de 5 elementos y
3   * accede a cada elemento del arreglo a través de un ciclo for.
4   */
5  int main ()
6  {
7      char ao=162;
8      int enteroNumAlumnos = 5;
9      float realCalif = 0.0, realPromedio = 0.0;
10     printf("\tPromedio de calificaciones\n");
11     for (int indice = 0 ; indice < enteroNumAlumnos ; indice++)
12     {
13         printf("\nIngrese la calificación del alumno %d\n",ao, indice+1);
14         scanf("%f",&realCalif);
15         realPromedio += realCalif;
16     }
17     printf("\nEl promedio de las calificaciones ingresadas es: %f\n",
18         realPromedio/enteroNumAlumnos);
19     return 0;
20 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc codesrefor.c -o codesrefor.exe
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>codesrefor.exe
Promedio de calificaciones
```

```
Ingrese la calificación del alumno 1
9
```

```
Ingrese la calificación del alumno 2
5
```

```
Ingrese la calificación del alumno 3
3
```

```
Ingrese la calificación del alumno 4
10
```

```
Ingrese la calificación del alumno 5
9
```

```
El promedio de las calificaciones ingresadas es: 7.200000
```



## Define

### Código (define)

```
codefine.c
1  #include <stdio.h>
2  #define MAX 5
3  /*
4   * Este programa define un valor por defecto para el tamaño del arreglo
5   * de tal manera que si el tamaño de éste cambia, solo se debe modificar
6   * el valor de la constante MAX.
7   */
8  int main ()
9  {
10     int arreglo[MAX], cont;
11     for (cont=0; cont<MAX; cont++)
12     {
13         printf("Ingrese el valor %d del arreglo: ", cont+1);
14         scanf("%i", &arreglo[cont]);
15     }
16     printf("El valor ingresado para cada elemento del arreglo es:\n[\\t");
17     for (cont=0; cont<MAX; cont++)
18     {
19         printf("%d\\t", arreglo[cont]);
20     }
21     printf("]\\n");
22     return 0;
23 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc codefine.c -o codefine.exe
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>codefine.exe
```

```
Ingrese el valor 1 del arreglo: 3
```

```
Ingrese el valor 2 del arreglo: 23
```

```
Ingrese el valor 3 del arreglo: 21
```

```
Ingrese el valor 4 del arreglo: 43
```

```
Ingrese el valor 5 del arreglo: 4
```

```
El valor ingresado para cada elemento del arreglo es:
```

```
[      3      23      21      43      4      ]
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```

## Break

### Código (break)

```
codefine.c cobreak.c
1  #include <stdio.h>
2  /*
3   * Este programa hace una suma de números. Si la suma rebasa la cantidad
4   * de 50 el programa se detiene.
5   */
6  #define VALOR_MAX 5
7  int main ()
8  {
9      char au=163;
10     char ao=162;
11     char ai=161;
12     int enteroSuma = 0;
13     int enteroNumero = 0;
14     int enteroContador = 0;
15     while (enteroContador < VALOR_MAX)
16     {
17         printf("\nIngrese un número:",au);
18         scanf("%d", &enteroNumero);
19         enteroSuma += enteroNumero;
20         enteroContador++;
21         if (enteroSuma > 50)
22         {
23             printf("Se rebasó la cantidad límite.\n",ao,ai);
24             break;
25         }
26     }
27     printf("\nEl valor de la suma es: %d\n", enteroSuma);
28     return 0;
29 }
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cobreak.c -o cobreak.exe
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cobreak.exe
```

```
Ingrese un número:1
```

```
Ingrese un número:2
```

```
Ingrese un número:3
```

```
Ingrese un número:4
```

```
Ingrese un número:5
```

```
El valor de la suma es: 15
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>cobreak.exe
```

```
Ingrese un número:10
```

```
Ingrese un número:20
```

```
Ingrese un número:30
```

```
Se rebasó la cantidad límite.
```

```
El valor de la suma es: 60
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```



## Continue

### Código (continue)

```
codefine.c x cobreak.c x cocon.c x
1  #include <stdio.h>
2  /*
3   * Este programa obtiene la suma de un LIMITE de números pares ingresados
4   * */
5  #define LIMITE 5
6  int main ()
7  {
8      char au="163";
9      int enteroContador = 1;
10     int enteroNumero = 0;
11     int enteroSuma = 0;
12     while (enteroContador <= LIMITE)
13     {
14         printf("Ingrese n%cmnero par %d:",au, enteroContador);
15         scanf("%d",&enteroNumero);
16         if (enteroNumero%2 != 0)
17         {
18             printf("El n%cmnero insertado no es par.\n",au);
19             continue;
20         }
21         enteroSuma += enteroNumero;
22         enteroContador++;
23     }
24     printf("La suma de los n%cmmeros es: %d\n",au, enteroSuma);
25     return 0;
26 }
27
```

```
C:\Users\user\Desktop\Lenguaje C\Ejemplos>gcc cocon.c -o cocon.exe

C:\Users\user\Desktop\Lenguaje C\Ejemplos>cocon.exe
Ingrese número par 1:1
El número insertado no es par.
Ingrese número par 1:2
Ingrese número par 2:4
Ingrese número par 3:6
Ingrese número par 4:12
Ingrese número par 5:24
La suma de los números es: 48

C:\Users\user\Desktop\Lenguaje C\Ejemplos>
```

### Conclusión:

En conclusión se aprendió y se elaboraron programas que ocupen las estructuras de repetición (while, do-while y for) para crear estructuras cíclicas, y que su uso es exclusivamente para ello, siempre y cuando que se ocupe la expresión lógica a evaluar. Además se aprendió a definir un símbolo ya anteriormente definido con otra cosa como lo es texto, se aprendió el break, que se ocupa para salir de un ciclo y el continue es sirve para provocar que inicie lo que sigue después del ciclo de repetición que lo contiene.