



Estructuras de repetición

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M. I. Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No. de Práctica(s): 9

Integrante(s): Sánchez García Rocío

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

No. de Lista: 47

Semestre: 2021-1

Fecha de entrega: Domingo 6 de diciembre de 2020

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*.

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*.
- ❖ Usar la directiva *define* para elaboración de código versátil.

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

Estructura de control repetitiva *while*

En esta estructura primero se valida la expresión lógica y de ser verdadera, ejecuta el bloque de instrucciones de la estructura. Si la condición resulta ser falsa se continua con el flujo normal del programa.

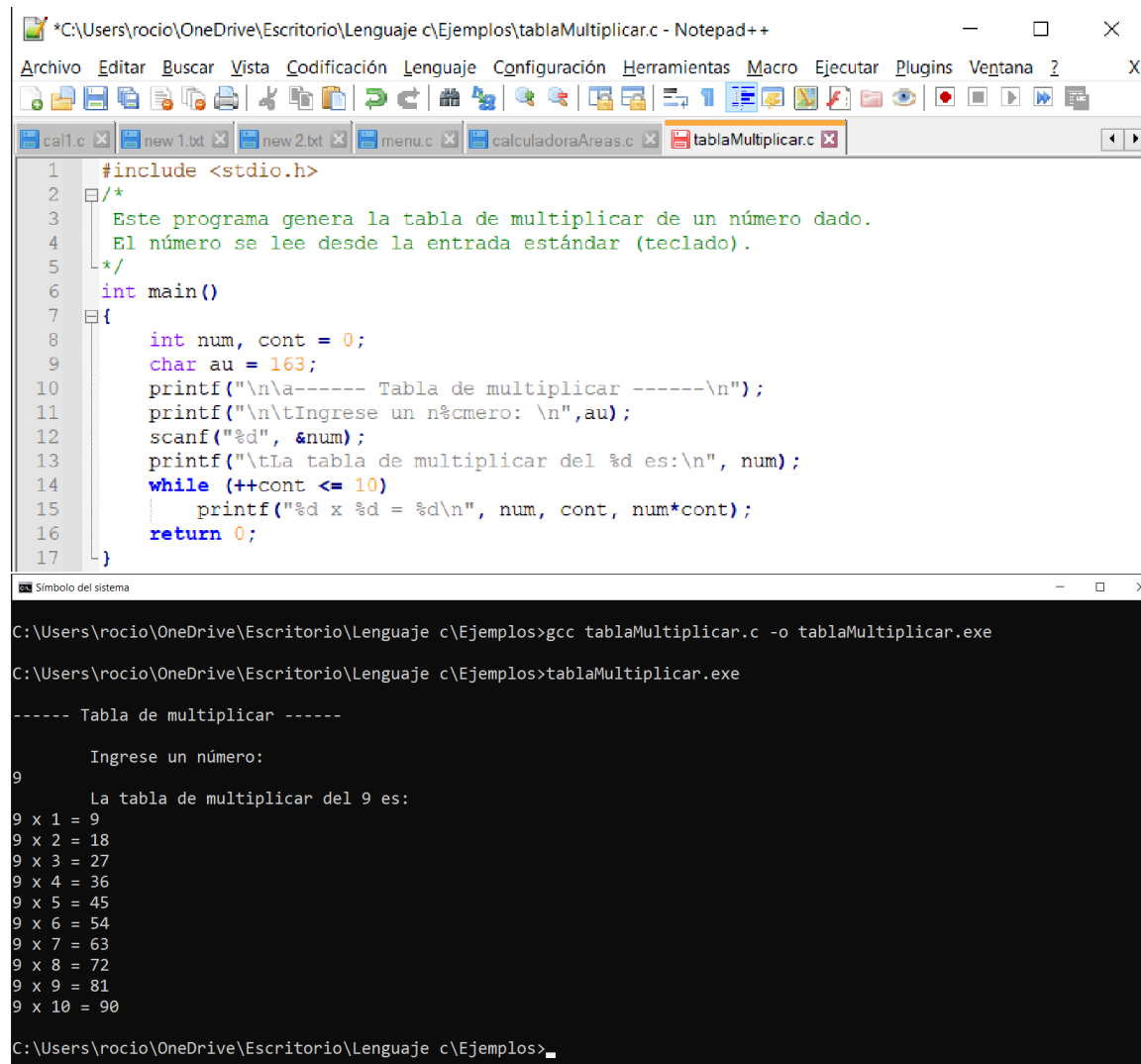
*El bloque se puede ejecutar las veces que sea necesario.

*Si el bloque de código consta de una sola sentencia, entonces se pueden omitir las llaves.

Sintaxis:

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

Código (estructura de repetición while)

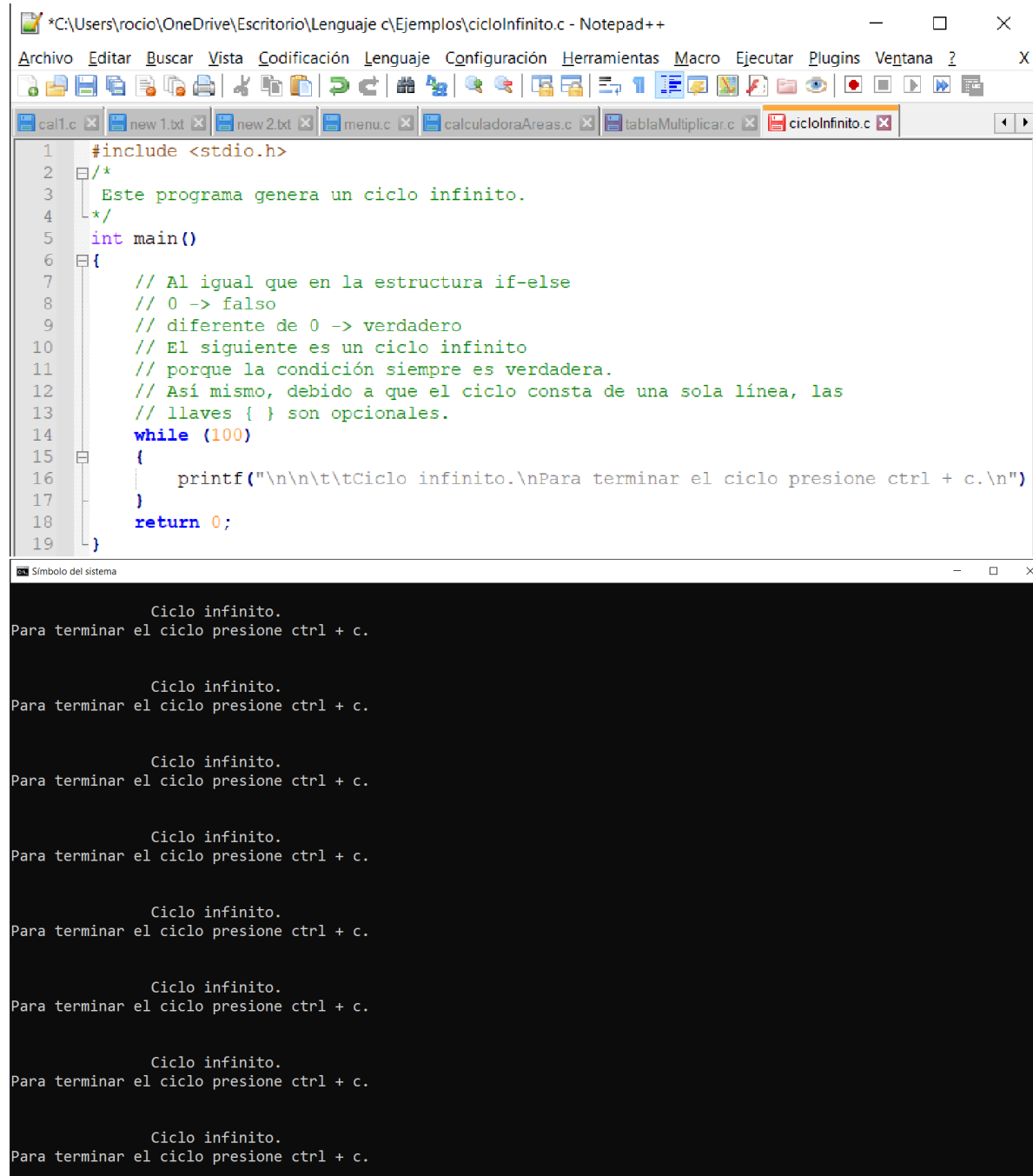


The image shows a Notepad++ window with a C program named `tablaMultiplicar.c`. The program generates a multiplication table for a user-input number. Below the code editor, a Windows command prompt shows the compilation and execution of the program, with the output displaying the multiplication table for the number 9.

```
*C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos\tablaMultiplicar.c - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ? X
cal1.c new 1.txt new 2.txt menu.c calculadoraAreas.c tablaMultiplicar.c
1 #include <stdio.h>
2 /*
3  Este programa genera la tabla de multiplicar de un número dado.
4  El número se lee desde la entrada estándar (teclado).
5  */
6 int main()
7 {
8     int num, cont = 0;
9     char au = 163;
10    printf("\n\a----- Tabla de multiplicar ----- \n");
11    printf("\n\tIngrese un número: \n", au);
12    scanf("%d", &num);
13    printf("\tLa tabla de multiplicar del %d es:\n", num);
14    while (++cont <= 10)
15        printf("%d x %d = %d\n", num, cont, num*cont);
16    return 0;
17 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc tablaMultiplicar.c -o tablaMultiplicar.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>tablaMultiplicar.exe
----- Tabla de multiplicar -----
      Ingrese un número:
9
      La tabla de multiplicar del 9 es:
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Código (estructura de repetición while)



```
*C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos\cicloInfinito.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X

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     // El siguiente es un ciclo infinito
11     // porque la condición siempre es verdadera.
12     // Así mismo, debido a que el ciclo consta de una sola línea, las
13     // llaves { } son opcionales.
14     while (100)
15     {
16         printf("\n\n\t\tCiclo infinito.\nPara terminar el ciclo presione ctrl + c.\n")
17     }
18     return 0;
19 }
```

```
Simbolo del sistema

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.
```

Estructura de control repetitiva do-while

do-while ejecuta el bloque de código que se encuentra en la estructura y después valida la condición.

*El Bloque de código se ejecuta la cantidad de veces que sea necesaria

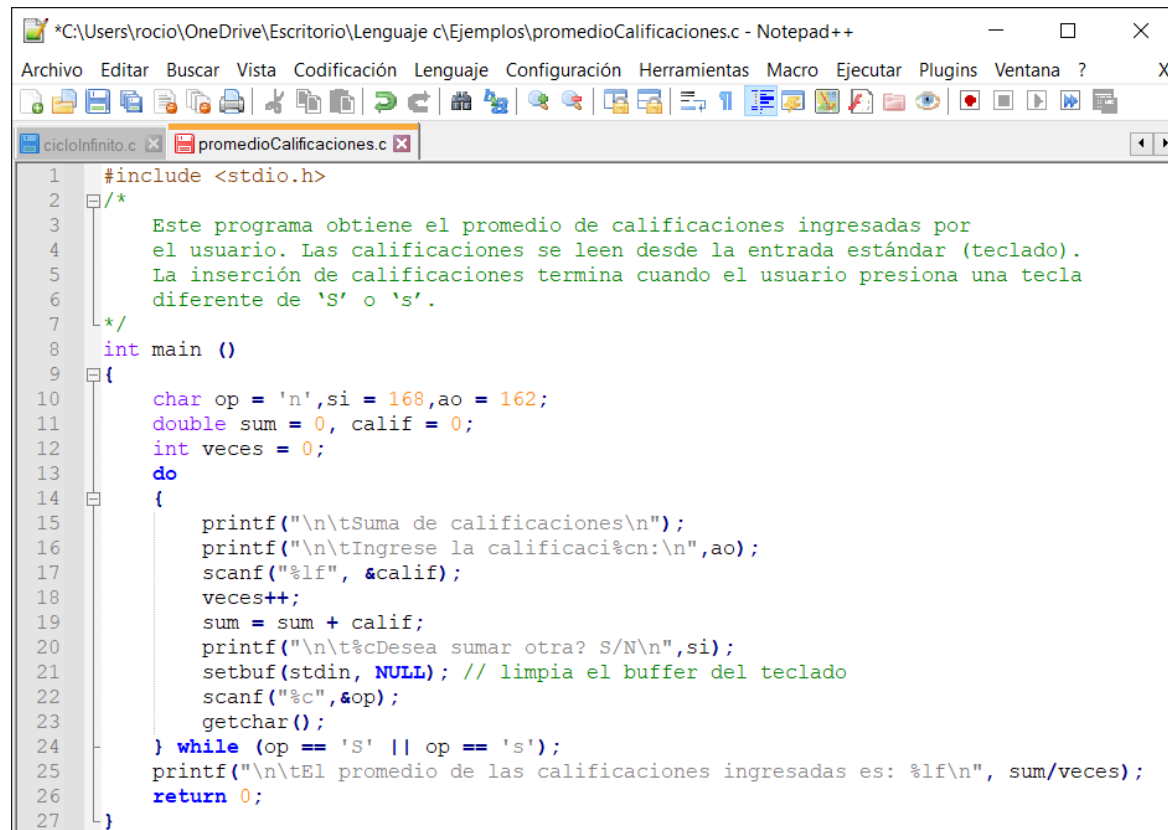
*Si el bloque de código consta de una sola sentencia entonces se pueden omitir las llaves.

*Esta estructura de control siempre termina con el signo de puntuación";".

Sintaxis:

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

Código (estructura de repetición do-while)



```
*C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos\promedioCalificaciones.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?  X

cicloInfinito.c  promedioCalificaciones.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 op = 'n', si = 168, ao = 162;
11     double sum = 0, calif = 0;
12     int veces = 0;
13     do
14     {
15         printf("\n\tSuma de calificaciones\n");
16         printf("\n\tIngrese la calificación: ", ao);
17         scanf("%lf", &calif);
18         veces++;
19         sum = sum + calif;
20         printf("\n\t¿Desea sumar otra? S/N\n", si);
21         setbuf(stdin, NULL); // limpia el buffer del teclado
22         scanf("%c", &op);
23         getchar();
24     } while (op == 'S' || op == 's');
25     printf("\n\tEl promedio de las calificaciones ingresadas es: %lf\n", sum/veces);
26     return 0;
27 }
```

```
Símbolo del sistema
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc promedioCalificaciones.c -o promedioCalificaciones.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>promedioCalificaciones.exe

Suma de calificaciones
Ingrese la calificación:
10

¿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:
8

¿Desea sumar otra? S/N
n

El promedio de las calificaciones ingresadas es: 9.000000
```

Código (estructuras de repetición do-while)

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos\calculadoraBasica.c - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
cicloInfinito.c promedioCalificaciones.c calculadoraBasica.c
1 #include <stdio.h>
2 /* Este programa genera una calculadora básica. */
3 int main ()
4 {
5     char si = 168, aa = 160, ae = 130, ao = 162, au = 163;
6     int op, uno, dos;
7     do
8     {
9         printf("\n\t ---- Calculadora ----\n");
10        printf("\n\t%c%cQu%c desea hacer\n", si, ae);
11        printf("\t1) Sumar\n");
12        printf("\t2) Restar\n");
13        printf("\t3) Multiplicar\n");
14        printf("\t4) Dividir\n");
15        printf("\t5) Salir\n");
16        scanf("%d", &op);
17        switch(op)
18        {
19            case 1:
20                printf("\tSumar\n");
21                printf("\tIntroduzca los números a sumar separados por comas\n", au);
22                scanf("%d, %d", &uno, &dos);
23                printf("\t%d + %d = %d\n", uno, dos, (uno + dos));
24                break;
25            case 2:
26                printf("\tRestar\n");
27                printf("\tIntroduzca los números a restar separados por comas\n", au);
28                scanf("%d, %d", &uno, &dos);
29                printf("\t%d - %d = %d\n", uno, dos, (uno - dos));
30                break;
31            case 3:
32                printf("\tMultiplicar\n");
33                printf("\tIntroduzca los números a multiplicar separados por comas\n", au);
34                scanf("%d, %d", &uno, &dos);
35                printf("\t%d * %d = %d\n", uno, dos, (uno * dos));
36                break;
37            case 4:
38                printf("\tDividir\n");
39                printf("\tIntroduzca los números a dividir separados por comas\n", au);
40                scanf("%d, %d", &uno, &dos);
41                printf("\t%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
42                break;
43            case 5:
44                printf("\tSalir\n");
45                break;
46            default:
47                printf("\tOpción inválida.\n", ao, aa);
48        }
49    }
50    while (op != 5);
51    return 0;
52 }
```

```
Símbolo del sistema - calculadoraBasica.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc calculadoraBasica.c -o calculadoraBasica.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>calculadoraBasica.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
10,9
    10 + 9 = 19

    ---- Calculadora ----

    ¿Qué desea hacer
    1) Sumar
    2) Restar
    3) Multiplicar
    4) Dividir
    5) Salir
2
    Restar
    Introduzca los números a restar separados por comas
10,9
    10 - 9 = 1

    ---- Calculadora ----

    ¿Qué desea hacer
    1) Sumar
    2) Restar
    3) Multiplicar
    4) Dividir
    5) Salir
3
    Multiplicar
    Introduzca los números a multiplicar separados por comas
10,9
    10 * 9 = 90

    ---- Calculadora ----

    ¿Qué desea hacer
    1) Sumar
    2) Restar
    3) Multiplicar
    4) Dividir
    5) Salir
4
    Dividir
    Introduzca los números a dividir separados por comas
10,9
    10 / 9 = 1.11

    ---- Calculadora ----

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

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Estructura de control de repetición *for*

Permite realizar repeticiones cuando se conoce el número de elementos que se quiere recorrer.

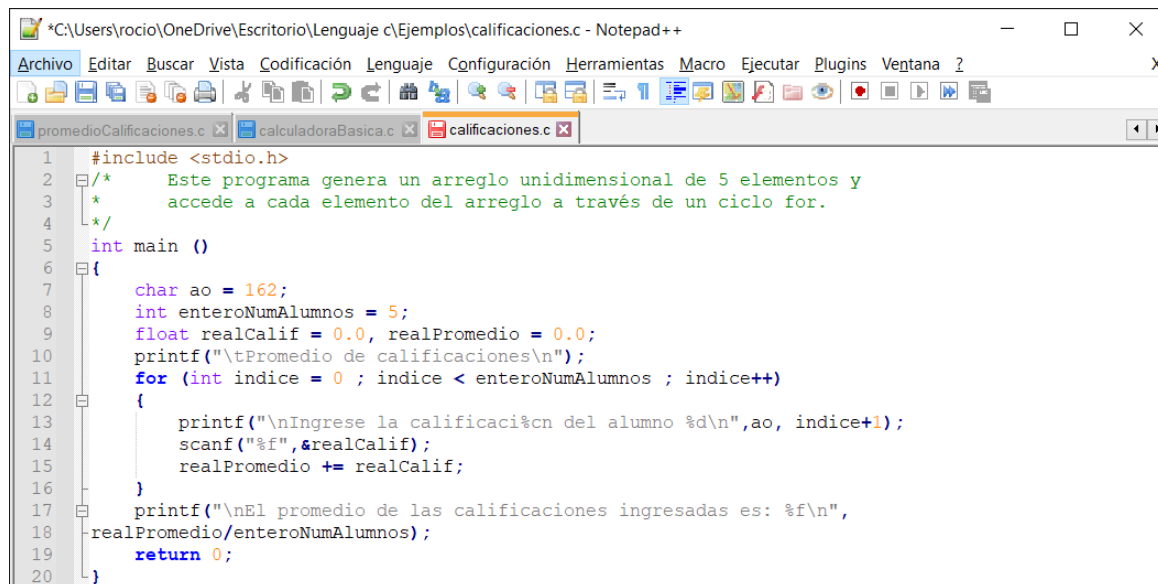
Ejecuta 3 acciones básicas antes o después de ejecutar el bloque de código.

- ❖ La primera acción es la inicialización, en la cual se pueden definir variables e inicializar sus valores, solo se ejecuta una vez cuando se ingresa al ciclo y es opcional.
- ❖ La segunda acción consta de una expresión lógica, la cual se evalúa y de ser verdadera ejecuta el bloque de código, de lo contrario se continua con la ejecución del programa; esta parte es opcional.
- ❖ La tercera parte consta de un conjunto de operaciones que se realizan cada vez que termina de ejecutarse el bloque de código y antes de volver a validar la expresión lógica; esta parte también es opcional.

Sintaxis:

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

Código (estructura de repetición for)



```
*C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos\calificaciones.c - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
promedioCalificaciones.c calculadoraBasica.c calificaciones.c
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 }
```

```
Símbolo del sistema
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc calificaciones.c -o calificaciones.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>calificaciones.exe
Promedio de calificaciones

Ingrese la calificación del alumno 1
8

Ingrese la calificación del alumno 2
7

Ingrese la calificación del alumno 3
7

Ingrese la calificación del alumno 4
9

Ingrese la calificación del alumno 5
10

El promedio de las calificaciones ingresadas es: 8.200000
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Define

Las líneas de código que empiezan por # son directivas del procesador, el nunca se encarga de realizar modificaciones en el texto del código fuente.

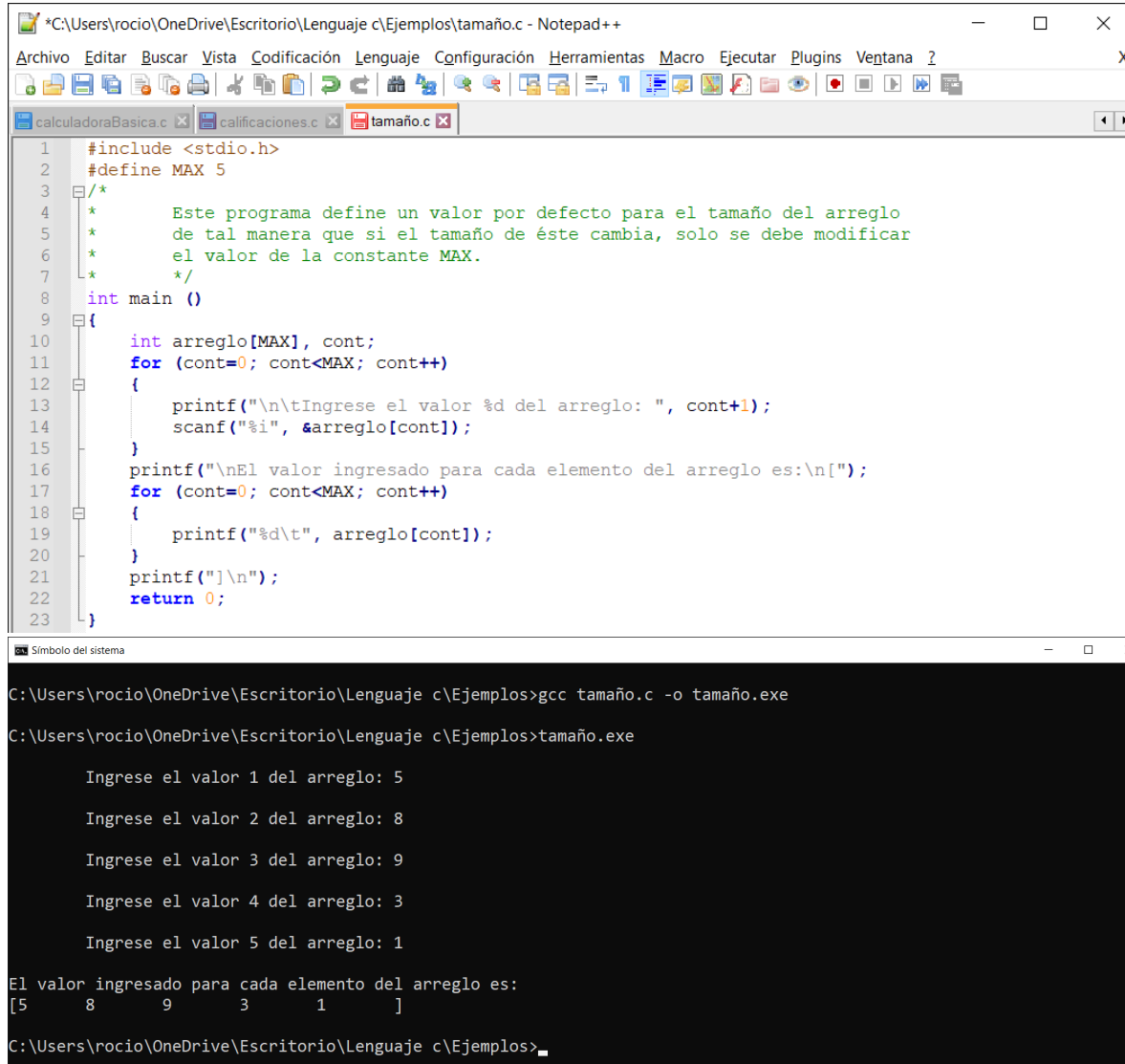
Define permite definir constantes o literales, se les nombra también como constantes simbólicas.

Sintaxis:

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

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

Código (define)



```
*C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos\tamaño.c - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
calculadoraBasica.c  calificaciones.c  tamaño.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("\n\tIngrese el valor %d del arreglo: ", cont+1);
14         scanf("%i", &arreglo[cont]);
15     }
16     printf("\nEl valor ingresado para cada elemento del arreglo es:\n[");
17     for (cont=0; cont<MAX; cont++)
18     {
19         printf("%d\t", arreglo[cont]);
20     }
21     printf("]\n");
22     return 0;
23 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc tamaño.c -o tamaño.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>tamaño.exe

Ingrese el valor 1 del arreglo: 5
Ingrese el valor 2 del arreglo: 8
Ingrese el valor 3 del arreglo: 9
Ingrese el valor 4 del arreglo: 3
Ingrese el valor 5 del arreglo: 1

El valor ingresado para cada elemento del arreglo es:
[5      8      9      3      1      ]

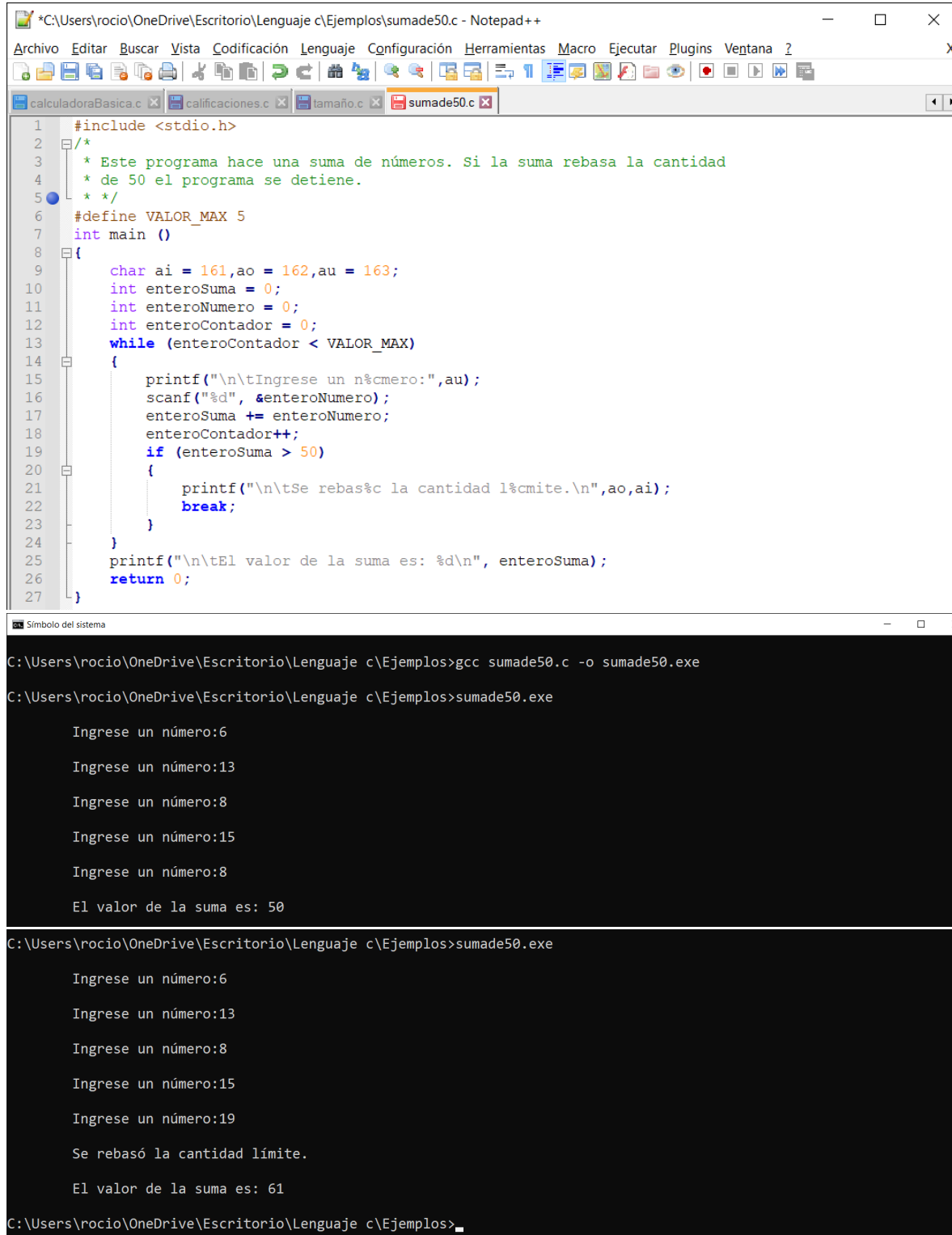
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_
```

Cuando se compila el programa, se reemplaza la palabra MAX por el valor definido para la misma. Esto permite que, si el tamaño del arreglo cambia, solo se tiene que modificar el valor definido para MAX y en automático todos los arreglos y el recorrido de los mismos adquieren un nuevo valor.

Break

Proporciona una salida anticipada dentro de una estructura de repetición, tal como lo hace en un *switch*. Un *break* provoca que el ciclo que lo encierra termine inmediatamente.

Código (break)



The image shows a Notepad++ window with a C program named `sumade50.c`. The program uses a `while` loop to sum numbers until the sum exceeds 50. A `break` statement is used to exit the loop when the limit is reached. Below the code editor, a Windows command prompt shows the compilation and execution of the program.

```
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 ai = 161,ao = 162,au = 163;
10    int enteroSuma = 0;
11    int enteroNumero = 0;
12    int enteroContador = 0;
13    while (enteroContador < VALOR_MAX)
14    {
15        printf("\n\tIngrese un n°cmero:",au);
16        scanf("%d", &enteroNumero);
17        enteroSuma += enteroNumero;
18        enteroContador++;
19        if (enteroSuma > 50)
20        {
21            printf("\n\tSe rebasó la cantidad límite.\n",ao,ai);
22            break;
23        }
24    }
25    printf("\n\tEl valor de la suma es: %d\n", enteroSuma);
26    return 0;
27 }
```

Command Prompt Output:

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc sumade50.c -o sumade50.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>sumade50.exe

Ingrese un número:6

Ingrese un número:13

Ingrese un número:8

Ingrese un número:15

Ingrese un número:8

El valor de la suma es: 50

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>sumade50.exe

Ingrese un número:6

Ingrese un número:13

Ingrese un número:8

Ingrese un número:15

Ingrese un número:19

Se rebasó la cantidad límite.

El valor de la suma es: 61

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Cuando se compila el programa, MAX se sustituye por 5.

Continue

La proposición *continue* provoca que inicie la siguiente iteración del ciclo de repetición que la contiene.

Código (continue)

```
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("\n\tIngrese n%cmero par %d: ",au, enteroContador);
15         scanf("%d",&enteroNumero);
16         if (enteroNumero%2 != 0)
17         {
18             printf("\n\tEl n%cmero insertado no es par.\n",auc);
19             continue;
20         }
21         enteroSuma += enteroNumero;
22         enteroContador++;
23     }
24     printf("\n\tLa suma de los n%cmeros es: %d\n",au, enteroSuma);
25     return 0;
26 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc sumaLimite.c -o sumaLimite.exe
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>sumaLimite.exe
```

```
    Ingrese número par 1: 4
```

```
    Ingrese número par 2: 8
```

```
    Ingrese número par 3: 6
```

```
    Ingrese número par 4: 2
```

```
    Ingrese número par 5: 10
```

```
    La suma de los números es: 30
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc sumalimite.c -o sumalimite.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>sumalimite.exe

Ingrese número par 1: 2
Ingrese número par 2: 6
Ingrese número par 3: 8
Ingrese número par 4: 9
El número insertado no es par.
Ingrese número par 4: 12
Ingrese número par 5: 2
La suma de los números es: 30
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

➤ Actividades asignadas por el profesor

Estructuras de repetición

Estas estructuras nos permiten repetir una o más instrucciones, ya sea un número determinado de veces o mientras se cumpla una condición.

- ❖ for
- ❖ while
- ❖ do-while

Ciclo for

Nos permite repetir una o más instrucciones un determinado número de veces.

Sintaxis:

```
for(inicialización; expresión_lógica; operaciones_por_iteración)
{
    //Instrucciones a repetir
}
```

Ejemplo:

Sumar los primeros 100 números como lo hizo Gauss pero utilizando el ciclo *for* en Lenguaje C.

```

1  #include<stdio.h>
2  int main()
3  {
4      //Declarar las variables
5      char au = 163, sp = 168, aa = 160;
6      int n, res;
7      //Mensaje de bienvenida
8      printf("\n\tSuma de los primeros n números\n\n", au);
9      //Solicitar el número de elementos a solicitar
10     printf("\t¿Cuántos números deseas sumar? ", sp, aa, au);
11     scanf("%d", &n);
12     //Sumar los n números
13     res = 0;
14     for(int i=1; i<=n; i++)
15     {
16         res = res+i;
17     }
18     //Mostrar el resultado
19     printf("\n\tLa suma de los primeros %d números es: %d\n", n, au, res);
20     return 0;
21 }

```

```
C:\Users\roocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc gauss.c -o gauss.exe
```

```
C:\Users\roocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gauss.exe
```

```
Suma de los primeros n números
```

```
¿Cuántos números deseas sumar? 100
```

```
La suma de los primeros 100 números es: 5050
```

```
C:\Users\roocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gauss.exe
```

```
Suma de los primeros n números
```

```
¿Cuántos números deseas sumar? 5
```

```
La suma de los primeros 5 números es: 15
```

```
C:\Users\roocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Ejercicio 1

Realizar un programa que calcule el factorial de un número.

$$1! = 1$$

$$2! = 2(1) = 2$$

$$3! = 3(2)(1) = 6$$

$$4! = 4(3)(2)(1) = 24$$

$$5! = 5(4)(3)(2)(1) = 120$$

```

1  #include<stdio.h>
2  int main()
3  {
4      //Declarar las variables
5      char au = 163;
6      int n,fact = 1;
7      //Mensaje de bienvenida
8      printf("\n\tFactorial de un n%cmero\n\n",au);
9      //Solicitar un número
10     printf("\tDigita el n%cmero: ",au);
11     scanf("%d",&n);
12     //Multiplicar los números
13     fact = 1;
14     for(int i=1;i<=n;i++)
15     {
16         fact = fact*i;
17     }
18     //Mostrar el resultado
19     printf("\n\tEl factorial del n%cmero %d es: %d \n",au,n,fact);
20     return 0;
21 }

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc factorial.c -o factorial.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>factorial.exe

Factorial de un número

Digita el número: 1

El factorial del número 1 es: 1

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>factorial.exe

Factorial de un número

Digita el número: 2

El factorial del número 2 es: 2

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>factorial.exe

Factorial de un número

Digita el número: 3

El factorial del número 3 es: 6

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>factorial.exe

Factorial de un número

Digita el número: 4

El factorial del número 4 es: 24

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>factorial.exe

Factorial de un número

Digita el número: 5

El factorial del número 5 es: 120

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>

Factorial de un número

Digita el número: 8

El factorial del número 8 es: 40320

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>

Ciclo while

Esta estructura nos permite validar una condición antes de realizar el ciclo.

Sintaxis:

```
while(condición o condiciones)
{
    //Instrucciones a repetir
}
```

Ejemplo:

Sumar los primeros 100 números como lo hizo Gauss pero ahora utilizando el ciclo *while* para hacer la comparativa.

```
1  #include<stdio.h>
2  int main()
3  {
4      //Declarar las variables
5      char au = 163, sp = 168, aa = 160;
6      int n, res, i;
7      //Mensaje de bienvenida
8      printf("\n\tSuma de los primeros n nmeros\n\n", au);
9      //Solicitar el nmero de elementos a solicitar
10     printf("\tCuntos nmeros deseas sumar? ", sp, aa, au);
11     scanf("%d", &n);
12     //Sumar los n nmeros
13     res = 0;
14     i = 1;
15     while(i<=n)
16     {
17         res = res+i;
18         i++;
19     }
20     //Mostrar el resultado
21     printf("\n\tLa suma de los primeros %d nmeros es: %d \n", n, au, res);
22     return 0;
23 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc gaussWhile.c -o gaussWhile.exe
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gaussWhile.exe
```

```
Suma de los primeros n nmeros
```

```
Cuntos nmeros deseas sumar? 100
```

```
La suma de los primeros 100 nmeros es: 5050
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Ejercicio 2

Convertir el programa del factorial ahora utilizando el ciclo *while*.

```
1  #include<stdio.h>
2  int main()
3  {
4      //Declarar las variables
5      char au = 163;
6      int n,fact = 1,i;
7      //Mensaje de bienvenida
8      printf("\n\tFactorial de un n%cmero\n\n",au);
9      //Solicitar un número
10     printf("\tDigita el n%cmero: ",au);
11     scanf("%d",&n);
12     //Multiplicar los números
13     fact = 1;
14     i = 1;
15     while(i<=n)
16     {
17         fact = fact*i;
18         i++;
19     }
20     //Mostrar el resultado
21     printf("\n\tEl factorial del n%cmero %d es: %d \n",au,n,fact);
22     return 0;
23 }
```

```
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc factorialWhile.c -o factorialWhile.exe
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>factorialWhile.exe

Factorial de un número

Digita el número: 5

El factorial del número 5 es: 120
C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_
```

Ciclo do-while

Es muy utilizado para la generación de menús, ya que nos permite repetirlo tantas veces como queramos mientras no se deje la opción salir. Y algo peculiar es que siempre se ejecuta al menos una vez.

Sintaxis:

Do

```
{
    //Instrucciones a repetir
} while(condición o instrucciones);
```

Ejemplo:

Programar un menú con opción de salida para repetirlo hasta que demos la opción salir.

```

1  #include<stdio.h>
2  int main()
3  {
4      //Declarar las variables
5      char au = 163,sp = 168,aa = 160,au = 163;
6      int n,res,i;
7      //Mensaje de bienvenida
8      printf("\n\tSuma de los primeros n n%cmeros\n\n",au);
9      //Solicitar el número de elementos a solicitar
10     printf("\t%cCu%cntos n%cmeros deseas sumar? ",sp,aa,au);
11     scanf("%d",&n);
12     //Sumar los n números
13     res = 0;
14     i = 1;
15     while(i<=n)
16     {
17         res = res+i;
18         i++;
19     }
20     //Mostrar el resultado
21     printf("\n\tLa suma de los primeros %d n%cmeros es: %d \n",n,au,res);
22     return 0;
23 }

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc menuSalir.c -o menuSalir.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>menuSalir.exe

Menú de figuras

- 1) Triángulo
- 2) Rectángulo
- 3) Círculo
- 4) Salir

Elige una opción: 1

Elegiste el Triángulo

- 1) Triángulo
- 2) Rectángulo
- 3) Círculo
- 4) Salir

Elige una opción: 2

Elegiste el Rectángulo

- 1) Triángulo
- 2) Rectángulo
- 3) Círculo
- 4) Salir

Elige una opción: 3

Elegiste el Círculo

```

1) Triángulo
2) Rectángulo
3) Círculo
4) Salir

Elige una opción: 5

Opción no válida!!!

1) Triángulo
2) Rectángulo
3) Círculo
4) Salir

Elige una opción: 4

Elegiste Salir
Gracias por usar nuestro programa

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>_

Tarea 5. Calculadora con opción de salir

Agrégle a un menú a la calculadora con la opción de salir, además del cálculo del factorial y la sumatoria de los primeros n números.

```

1 #include <stdio.h>
2 int main()
3 {
4     //Declarar las variables a utilizar
5     int n1,n2,n,res,i,fact=1;
6     char aa = 160,ao = 162,au = 163,mod = 37,op = '\0',sp = 168;
7     //Mensaje de bienvenida
8     printf("\n\n\t\t\tCalculadora\n\n");
9     do
10     {
11         //Mostrar menú
12         printf("\n\t1) Suma\n\t2) Resta\n\t3) Multiplicación\n\t4) División y Módulo\n\t5) Factorial\n\t6) Suma de los primeros n números\n\t7) Salir\n",ao,ao,ao,au);
13         //Solicitar la opción
14         printf("\n\tElija una operación: ",ao);
15         scanf("%d",&op);
16         switch(op)
17         {
18             case 1:
19                 //Solicitar las variables
20                 printf("\n\tIngrese dos números enteros separados por coma: ",au);
21                 scanf("%i,%i",&n1,&n2);
22                 printf("\n\t%d + %d = %d\n",n1,n2,n1+n2);
23                 break;
24             case 2:
25                 //Solicitar las variables
26                 printf("\n\tIngrese dos números enteros separados por coma: ",au);
27                 scanf("%i,%i",&n1,&n2);
28                 printf("\n\t%d - %d = %d\n",n1,n2,n1-n2);
29                 break;
30             case 3:
31                 //Solicitar las variables
32                 printf("\n\tIngrese dos números enteros separados por coma: ",au);
33                 scanf("%i,%i",&n1,&n2);
34                 printf("\n\t%d * %d = %d\n",n1,n2,n1*n2);
35                 break;
36             case 4:
37                 //Solicitar las variables
38                 printf("\n\tIngrese dos números enteros separados por coma: ",au);
39                 scanf("%i,%i",&n1,&n2);
40                 if (n1 > n2)
41                 {
42                     if (n2==0)
43                     {
44                         printf("\n\tIndeterminación\n",ao);
45                         printf("\n\tIngrese otro valor\n");
46                     }
47                     else
48                     {
49                         printf("\n\t%d / %d = %d\n",n1,n2,n1/n2);
50                         printf("\n\t%d %c %d = %d\n",n1,mod,n2,n1%n2);
51                     }
52                 }
53                 else
54                 {
55                     if (n2 > n1)
56                     {
57                         if (n1==0)
58                         {
59                             printf("\n\tIndeterminación\n",ao);
60                             printf("\n\tIngrese otro valor\n");
61                         }
62                         else
63                         {
64                             printf("\n\t%d / %d = %d\n",n2,n1,n2/n1);
65                             printf("\n\t%d %c %d = %d\n",n2,mod,n1,n2%n1);
66                         }
67                     }
68                 }
69                 break;
70             case 5:
71                 //Solicitar las variables
72                 printf("\n\tIngrese dos números enteros separados por coma: ",au);
73                 scanf("%i,%i",&n1,&n2);
74                 if (n1 > n2)
75                 {
76                     if (n2==0)
77                     {
78                         printf("\n\tIndeterminación\n",ao);
79                         printf("\n\tIngrese otro valor\n");
80                     }
81                     else
82                     {
83                         printf("\n\t%d / %d = %d\n",n1,n2,n1/n2);
84                         printf("\n\t%d %c %d = %d\n",n1,mod,n2,n1%n2);
85                     }
86                 }
87                 else
88                 {
89                     if (n2 > n1)
90                     {
91                         if (n1==0)
92                         {
93                             printf("\n\tIndeterminación\n",ao);
94                             printf("\n\tIngrese otro valor\n");
95                         }
96                         else
97                         {
98                             printf("\n\t%d / %d = %d\n",n2,n1,n2/n1);
99                             printf("\n\t%d %c %d = %d\n",n2,mod,n1,n2%n1);
100                        }
101                    }
102                }
103                break;
104            case 6:
105                //Solicitar las variables
106                printf("\n\tIngrese un número entero: ",au);
107                scanf("%i",&n);
108                fact=1;
109                for(i=1;i<=n;i++)
110                {
111                    fact=fact*i;
112                }
113                printf("\n\tEl factorial de %d es %d\n",n,fact);
114                break;
115            case 7:
116                //Solicitar las variables
117                printf("\n\tIngrese un número entero: ",au);
118                scanf("%i",&n);
119                res=0;
120                for(i=1;i<=n;i++)
121                {
122                    res=res+i;
123                }
124                printf("\n\tLa suma de los primeros %d números es %d\n",n,res);
125                break;
126            default:
127                printf("\n\tOpción no válida!!!\n");
128                continue;
129        }
130    } while (op != 7);
131    printf("\n\tGracias por usar nuestro programa\n");
132    return 0;
133 }

```

```

72 //Solicitar un número
73 printf("\n\tDigita el número: ",au);
74 scanf("%d",&n);
75 //Multiplicar los números
76 fact = 1;
77 i = 1;
78 while(i<=n)
79 {
80     fact = fact*i;
81     i++;
82 }
83 //Mostrar el resultado
84 printf("\n\tEl factorial del número %d es: %d \n",au,n,fact);
85 break;
86 case 6:
87     //Solicitar el número de elementos a solicitar
88     printf("\n\t¿Cuántos números deseas sumar? ",sp,aa,au);
89     scanf("%d",&n);
90     //Sumar los n números
91     res = 0;
92     i = 1;
93     while(i<=n)
94     {
95         res = res+i;
96         i++;
97     }
98     //Mostrar el resultado
99     printf("\n\tLa suma de los primeros %d números es: %d \n",n,au,res);
100     break;
101 case 7:
102     printf("\n\tHas salido de la calculadora\n");
103     break;
104 default:
105     printf("\n\tOpción no válida.\n",ao,aa);
106     break;
107 }
108 while(opt!=0);
109 return 0;
110 }

```

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>gcc calculadora5.c -o calculadora5.exe

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>calculadora5.exe

Calculadora

- 1) Suma
- 2) Resta
- 3) Multiplicación
- 4) División y Módulo
- 5) Factorial
- 6) Suma de los primeros n números
- 7) Salir

Elige una operación: 1

Ingresa dos números enteros separados por coma: 10,9

10 + 9 = 19

- 1) Suma
- 2) Resta
- 3) Multiplicación
- 4) División y Módulo
- 5) Factorial
- 6) Suma de los primeros n números
- 7) Salir

Elige una operación: 2

Ingresa dos números enteros separados por coma: 19,9

19 - 9 = 10

- 1) Suma
- 2) Resta
- 3) Multiplicación
- 4) División y Módulo
- 5) Factorial
- 6) Suma de los primeros n números
- 7) Salir

Elige una operación: 3

Ingresa dos números enteros separados por coma: 10,9

10 * 9 = 90

- 1) Suma
- 2) Resta
- 3) Multiplicación
- 4) División y Módulo
- 5) Factorial
- 6) Suma de los primeros n números
- 7) Salir

Elige una operación: 4

Ingresa dos números enteros separados por coma: 10,9

10 / 9 = 1

10 % 9 = 1

```
1) Suma
2) Resta
3) Multiplicación
4) División y Módulo
5) Factorial
6) Suma de los primeros n números
7) Salir

Elige una operación: 5

Digita el número: 5

El factorial del número 5 es: 120

1) Suma
2) Resta
3) Multiplicación
4) División y Módulo
5) Factorial
6) Suma de los primeros n números
7) Salir

Elige una operación: 6

¿Cuántos números deseas sumar? 10

La suma de los primeros 10 nmeros es: 55

1) Suma
2) Resta
3) Multiplicación
4) División y Módulo
5) Factorial
6) Suma de los primeros n números
7) Salir

Elige una operación: 7

Has salido de la calculadora

C:\Users\rocio\OneDrive\Escritorio\Lenguaje c\Ejemplos>
```

Conclusiones

- ❖ Gracias a esta estructura es posible ejecutar procesos ahorrándonos una gran cantidad de Código.
- ❖ Podemos aplicarlas en una gran variada de programas, por ejemplo, en la realización de un menú donde se tenga la necesidad de volver a mostrar una vez que se hallan terminado de ejecutar ciertas acciones dentro del programa.

Bibliografía

Manual de prácticas del Laboratorio de Fundamentos de Programación, Facultad de ingeniería UNAM, recuperada el 30 de noviembre, en <http://lcp02.fi-b.unam.mx/>