



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): 7

Integrante(s): López Martínez Diana

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 28

Semestre: 1

Fecha de entrega: 06/12/2020

Observaciones:

CALIFICACIÓN: _____

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.

Estructura de control repetitiva while

La estructura repetitiva (o iterativa) while primero valida la expresión lógica y si ésta se cumple (es verdadera) procede a ejecutar el bloque de instrucciones de la estructura, el cual está delimitado por las llaves {}. Si la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura, es decir, el bloque se puede ejecutar de cero a ene veces. Su sintaxis es la siguiente:

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

Código (estructura de repetición while)



tablaM: Bloc de notas

Archivo Edición Formato Ver Ayuda

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

```
/*
```

```
Este programa genera la tabla de multiplicar de un número dado.  
El número se lee desde la entrada estándar (teclado).
```

```
*/
```

```
int main(){
```

```
int num, cont = 0;
```

```
char au=163;
```

```
printf("\n\n\t\t\tTabla de multiplicar \n\n");
```

```
printf("Ingrese un número: \n",au);
```

```
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;
```

```
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc tablaM.c -o tablaM.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>tablaM.exe
```

Tabla de multiplicar

Ingrese un número:

6

La tabla de multiplicar del 6 es:

6 x 1 = 6

6 x 2 = 12

6 x 3 = 18

6 x 4 = 24

6 x 5 = 30

6 x 6 = 36

6 x 7 = 42


6 x 8 = 48

6 x 9 = 54

6 x 10 = 60

Código (estructura de repetición while)



 cicloWh: Bloc de notas

[Archivo](#)
[Edición](#)
[Formato](#)
[Ver](#)
[Ayuda](#)

```
#include <stdio.h>
/*
    Este programa genera un ciclo infinito.
*/
int main(){
    // Al igual que en la estructura if-else
    // 0 -> falso
    // 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;
}
```

[illegible]


Estructura de control repetitiva do-while

do-while es una estructura cíclica que ejecuta el bloque de código que se encuentra dentro de las llaves y después valida la condición, es decir, el bloque de código se ejecuta de una a ene veces. Su sintaxis es la siguiente:

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

Si el bloque de código a repetir consta de una sola sentencia, entonces se pueden omitir las llaves. Esta estructura de control siempre termina con el signo de puntuación ';'.

Código (estructura de repetición do-while)

 doWhile: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>  
/*  
    Este programa obtiene el promedio de calificaciones ingresadas por  
    el usuario. Las calificaciones se leen desde la entrada estándar (teclado).  
    La inserción de calificaciones termina cuando el usuario presiona una tecla  
    diferente de 'S' o 's'.  
*/  
int main () {  
    char op = 'n';  
    char ao = 162;  
    char sp = 168;  
    double sum = 0, calif = 0;  
  
    int veces = 0;  
    do {  
        printf("\n\n\t\t\tSuma de calificaciones\n\n");  
        printf("Ingrese la calificación:\n",ao);  
        scanf("%lf", &calif);  
        veces++;  
        sum = sum + calif;  
  
        printf("\nDesea sumar otra? S/N\n",sp);  
        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;  
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc dowhile.c -o dowhile.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>dowhile.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

N

El promedio de las calificaciones ingresadas es: 9.500000

Código (estructura de repetición do-while)



doWhiled: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>
/* Este programa genera una calculadora básica. */
int main () {
    int op, uno, dos;
    char sp = 168;
    char ae = 130;
    char au = 163;
    char ao = 162;
    char ad = 160;
    do {
        printf("\n\n\t\t\tCalculadora \n\n");
        printf("\n%cQu%c desea hacer\n",sp,ae);
        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 n%cmeros a sumar separados por comas\n",au);
                scanf("%d, %d",&uno, &dos);
                printf("%d + %d = %d\n", uno, dos, (uno + dos));
                break;
            case 2:
```

```

        printf("\tRestar\n");
        printf("Introduzca los n%cmeros a restar separados por comas\n",au);
        scanf("%d, %d",&uno, &dos);
        printf("%d - %d = %d\n", uno, dos, (uno - dos));
        break;
    case 3:
        printf("\tMultiplicar\n");
        printf("Introduzca los n%cmeros a multiplicar separados por comas\n",au);
        scanf("%d, %d",&uno, &dos);
        printf("%d * %d = %d\n", uno, dos, (uno * dos));
        break;
    case 4:
        printf("\tDividir\n");
        printf("Introduzca los n%cmeros a dividir separados por comas\n",au);
        scanf("%d, %d",&uno, &dos);
        printf("%d / %d = %.2lf\n", uno, dos, ((double)uno / dos));
        break;
    case 5:
        printf("\tSalir\n");
        break;
    default:
        printf("\topci%c\n inv%clida.\n",ao,ad);
    }
} while (op != 5);

return 0;
}

```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc dowhiled.c -o dowhiled.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>dowhiled.exe
```

Calculadora

¿Qué desea hacer

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

2

Restar

Introduzca los números a restar separados por comas

78,50

78 - 50 = 28

Calculadora

¿Qué desea hacer

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

5

Salir

Estructura de control de repetición for

Lenguaje C posee la estructura de repetición for la cual permite realizar repeticiones cuando se conoce el número de elementos que se quiere recorrer. La sintaxis que generalmente se usa es la siguiente:

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

La estructura for 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; esta parte 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, si ésta es verdadera, ejecuta el bloque de código, si no se cumple se continúa 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.

Código (estructura de repetición for)

 for: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include <stdio.h>  
/* Este programa genera un arreglo unidimensional de 5 elementos y  
 * accede a cada elemento del arreglo a través de un ciclo for.  
 */  
int main () {  
    char ao= 162;  
    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 calificaci%cn del alumn %d\n",ao, indice+1);  
        scanf("%f",&realCalif);  
        realPromedio += realCalif;  
    }  
  
    printf("\nEl promedio de las calificaciones ingresadas es: %f\n",  
        realPromedio/enteroNumAlumnos);  
  
    return 0;  
}
```



```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>for.exe
```

```
                Promedio de calificaciones
```

```
Ingrese la calificación del alumn 1
```

```
8
```

```
Ingrese la calificación del alumn 2
```

```
10
```

```
Ingrese la calificación del alumn 3
```

```
9
```

```
Ingrese la calificación del alumn 4
```

```
6
```

```
Ingrese la calificación del alumn 5
```

```
2
```

```
El promedio de las calificaciones ingresadas es: 7.000000
```

Define

Las líneas de código que empiezan con # son directivas del preprocesador, el cual se encarga de realizar modificaciones en el texto del código fuente, como reemplazar un símbolo definido con #define por un parámetro o texto, o incluir un archivo en otro archivo con #include.

define permite definir constantes o literales; se les nombra también como constantes simbólicas. Su sintaxis es la siguiente:

#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 cambiará por el valor definido. El valor puede ser numérico o puede ser texto.

Código (define)

```
*define: Bloc de notas
Archivo Edición Formato Ver Ayuda
#include <stdio.h>
#define MAX 5

/*
 * Este programa define un valor por defecto para el tamaño del arreglo
 * de tal manera que si el tamaño de éste cambia, solo se debe modificar
 * el valor de la constante MAX.
 * */

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;
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc define.c -o define.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>define.exe
```

```
Ingrese el valor 1 del arreglo: 5
```

```
Ingrese el valor 2 del arreglo: 10
```

```
Ingrese el valor 3 del arreglo: 8
```

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

```
Ingrese el valor 5 del arreglo: 9
```


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

```
[5      10      8      5      9      ]
```

Break

Algunas veces es conveniente tener la posibilidad de abandonar un ciclo. La proposición 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)

 break: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
/*
 * Este programa hace una suma de números. Si la suma rebasa la cantidad
 * de 50 el programa se detiene.
 */
#define VALOR_MAX 5

int main (){
char au= 163;
char ao= 162;
char ai= 161;
    int enteroSuma = 0;
    int enteroNumero = 0;
    int enteroContador = 0;
    while (enteroContador < VALOR_MAX){
        printf("Ingrese un número:",au);
        scanf("%d", &enteroNumero);
        enteroSuma += enteroNumero;
        enteroContador++;
        if (enteroSuma > 50){
            printf("Se rebasó la cantidad límite.\n",ao,ai);
            break;
        }
    }
    printf("El valor de la suma es: %d\n", enteroSuma);

    return 0;
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc break.c -o break.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>break.exe
```

```
Ingrese un número:8
```

```
Ingrese un número:25
```

```
Ingrese un número:65
```

```
Se rebasó la cantidad límite.
```


```
El valor de la suma es: 98
```

Continue

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

contiene.

Código (continue)

 continue: Bloc de notas

Archivo Edición Formato Ver Ayuda

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

```
/*
```

```
 * Este programa obtiene la suma de un LIMITE de números pares ingresados
 * */
```

```
#define LIMITE 5
```

```
int main (){
```

```
char au = 163;
```

```
    int enteroContador = 1;
```

```
    int enteroNumero = 0;
```

```
    int enteroSuma = 0;
```

```
    while (enteroContador <= LIMITE){
```

```
        printf("Ingrese n%cmero par %d:",au, enteroContador);
```

```
        scanf("%d",&enteroNumero);
```

```
        if (enteroNumero%2 != 0){
```

```
            printf("El n%cmero insertado no es par.\n",au);
```

```
            continue;
```

```
        }
```

```
        enteroSuma += enteroNumero;
```

```
        enteroContador++;
```

```
    }
```

```
    printf("La suma de los n%cmeros es: %d\n",au, enteroSuma);
```

```
    return 0;
```

```
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc continue.c -o continue.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>continue.exe
```

```
Ingrese número par 1:2
```

```
Ingrese número par 2:6
```

```
Ingrese número par 3:8
```

```
Ingrese número par 4:4
```

```
Ingrese número par 5:12
```

```
La suma de los números es: 32
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc continue.c -o continue.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>continue.exe
```

```
Ingrese número par 1:6
```

```
Ingrese número par 2:7
```

```
El número insertado no es par.
```

```
Ingrese número par 2:8
```

```
Ingrese número par 3:9
```

```
El número insertado no es par.
```

```
Ingrese número par 3:12
```

```
Ingrese número par 4:10
```

```
Ingrese número par 5:4
```

```
La suma de los números es: 40
```

Estructura de Repetición

gaus: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include<stdio.h>
int main()
{
    //Declarar variables
    char sp =168;
    char au=163;
    char ad=160;
    int n,res,i;

    //Mensaje de Bienvenida
    printf("\n\n\t\t\tSuma de los primeros n n%cmeros\n\n",au);

    //Solicitar el numero de elementos a sumar
    printf("%cCu%cntos n%cmeros deseas sumar? ",sp,ad,au);
    scanf("%d",&n);

    //Sumar los n numeros
    res=0;
    for(int i=1;i<=n;i++)
    {
        res=res+i;
    }

    //Mostrar el resultado
    printf("La suma de los primeros %d n%cmeros es: %d \n",n,au,res);

    return 0;
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc gaus.c -o gaus.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gaus.exe
```

Suma de los primeros n números

¿Cuántos números deseas sumar? 100

La suma de los primeros 100 números es: 5050

Factorial



pFac: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include<stdio.h>
int main()
{
    //Declarar las variables
    char au =163;
    int i, n, res;

    //Mensaje de Bienvenida
    printf("\n\n\t\t\tCalculadora Factorial\n\n",au);

    //Solicitar el numero a calcular
    printf("introduce un n%cmero: \n",au);
    scanf("%d",&n);

    //Calcular el factorial
    res=0;
    for (int i=n;i>1;i++)
    {
        res=res*i;
    }

    //Mostrar el resultado
    printf("El factorial del n%cmero solicitado es: %d \n",n,au,res);
}
```

Tuve problemas al realizar el programa

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc pFac.c -o pFac.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>pFac.exe
```

Calculadora Factorial

introduce un número:

4

Ciclo While

*gaussWhile: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include<stdio.h>
int main()
{
    //Declarar variables
    char sp =168;
    char au=163;
    char ad=160;
    int n,res,i;

    //Mensaje de Bienvenida
    printf("\n\n\t\t\tSuma de los primeros n n%cmeros\n\n",au);

    //Solicitar el numero de elementos a sumar
    printf("%cCu%cntos n%cmeros deseas sumar?",sp,ad,au);
    scanf("%d",&n);

    //Sumar los n numeros
    res=0;
    i=1;
    while(i<=n)
    {
        res=res+1;
        i++;
    }

    //Mostrar el resultado
    printf("La suma de los primeros %d n%cmeros es: %d \n",n,au,res);

    return 0;
}
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc gaussWhile.c -o gaussWhile.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gaussWhile.exe
```

Suma de los primeros n números

¿Cuántos números deseas sumar? 100

La suma de los primeros 100 números es: 100

```
#include<stdio.h>
int main()
{
    //Declarar las variables
    char au =163;
    int i, n, res;

    //Mensaje de Bienvenida
    printf("\n\n\t\t\tCalculadora Factorial\n\n",au);

    //Solicitar el numero a calcular
    printf("introduce un n°mero: \n",au);
    scanf("%d",&n);

    //Calcular el factorial
    res=0;
    i=1;
    while(i>1)
    {
        res=res*i;
        i--;
    }

    //Mostrar el resultado
    printf("El factorial del n°mero solicitado es: %d \n",n,au,res);
}
```

C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc facWhile.c -o facWhile.exe

C:\Users\paust\Downloads\Lenguaje c\Ejemplos>facWhile.exe


Calculadora Factorial

introduce un número:

4

El factorial del n°mero solicitado es: -93

submenú

 subMenu: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#include<stdio.h>
int main()
{
    //Declarar variables
    int op,op2;
    char ad=160, ae=130, ai=161, ao=162, au=163;

    //Mensaje de bienvenida
    printf("\n\n\t\t\tMen%c de figuras :)\n\n",au);

    do
    {
        //Mostrar el menu
        printf("1) Tri%cngulo\n2) Rect%cngulo\n3) C%rculo\n4) Salir\n",ad,ad,ai);

        //Solicitar la opcion
        printf("Elige una opci%c: ",ao);
        scanf("%d",&op);

        switch(op)
        {
            case 1:
                printf("Elegiste Tri%cngulo\n\n",ad);
                do
                {
                    printf("1) %crea\n2) per%cmetro\n3) salir\n",ad,ai);
                    printf("Elige una opci%c: ",ao);
                    scanf("%d",&op2);
                    switch(op2)
                    {
                        case 1:
                            printf("Elegiste el %crea\n\n",ad);
                            break;
                        case 2:
                            printf("Elegiste par%cmetro\n\n,ad");
                            break;
                        case 3:
                            printf("Elegiste salir\n\n");
                            break;
                        default:
                            printf("Opci%c no v%clida!!!\n\n",ao,ad);
                    }
                }while(op2!=3);
                break;
            case 2:
                printf("Elegiste Rect%cngulo\n\n",ad);
                do
                {
                    printf("1) %crea\n2) per%cmetro\n3) salir\n",ad,ai);
                    printf("Elige una opci%c: ",ao);
                    scanf("%d",&op2);
                    switch(op2)
                    {
                        case 1:
                            printf("Elegiste el %crea\n\n",ad);
                            break;
```

```

        case 2:
            printf("Elegiste par%cmetro\n\n",ad");
            break;
        case 3:
            printf("Elegiste salir\n\n");
            break;
        default:
            printf("Opci%c\n no v%clida!!!\n\n",ao,ad);
    }
}while(op2!=3);
break;
case 3:
    printf("Elegiste C%crculo\n\n",ai);
    do
    {
        printf("1) %crea\n2) per%cmetro\n3) salir\n",ad,ai);
        printf("Elige una opci%c\n: ",ao);
        scanf("%d",&op2);
        switch(op2)
        {
            case 1:
                printf("Elegiste el %crea\n\n",ad);
                break;
            case 2:
                printf("Elegiste par%cmetro\n\n",ad");
                break;
            case 3:
                printf("Elegiste salir\n\n");
                break;
            default:
                printf("Opci%c\n no v%clida!!!\n\n",ao,ad);
        }
    }while(op2!=3);
    break;
case 4:
    printf("Elegiste Salir\n\n");
    default:
        printf("Opci%c\n no v%clida!!!\n\n",ao,ad);
}
}while(op!=4);
printf("Gracias por usar nuestro programa :)\n");
return 0;
}

```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>gcc subMenu.c -o subMenu.exe
```

```
C:\Users\paust\Downloads\Lenguaje c\Ejemplos>subMenu.exe
```

Menú de figuras :)

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

Elige una opción: 1

Elegiste Triángulo

- 1) área
- 2) perímetro
- 3) salir

Elige una opción: 1

Elegiste el área

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

Elige una opción: 3

Elegiste Círculo

- 1) área
- 2) perímetro
- 3) salir

Elige una opción: 1

Elegiste el área

- 1) área
- 2) perímetro
- 3) salir

Elige una opción: 2


Elegiste par▶metro

- ,ad1) área
- 2) perímetro
- 3) salir

Elige una opción: 3

Elegiste salir

Calculadora

 calFinal: Bloc de notas

Archivo Edición Formato Ver Ayuda

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

{
    char ao=162;
    //Mensaje de Bienvenida
    printf("\n\n\t\tBienvenido a mi Calculadora. :)\n\n");

    enum opeCal {SUMA, RESTA, MULTIPLICACION, DIVISION, MODULO};
    int op;
    printf("Por favor ingrese la operaci%cn que desea realizar.\n",ao);
    printf("1) SUMA\n");
    printf("2) RESTA\n");
    printf("3) MULTIPLICACI%cN\n",ao);
    printf("4) DIVICI%cN\n",ao);
    printf("5) M%cDULO\n",ao);
    printf("6) Factorial\n");
    printf("7) Gauss\n");
    printf("8) Salir\n");
    scanf("%d", &op);

    //Declarado variables a utilizar
    int n1,n2,res,n,i;
    char au=163;
    char sp=168;
    char ad=160;

    if(op==1)
    {
        //Solicitar variables
        printf("Dame 2 n%cmeros separados por coma: ",au);
        scanf("%i,%i",&n1,&n2);

        //Realizar la suma
        res=n1+n2;
        printf("La suma de %d y %d es: %d \n",n1,n2,res);

    }
    else if(op==2)
    {
        //Solicitar variables
        printf("Dame 2 n%cmeros separados por coma: ",au);
        scanf("%i,%i",&n1,&n2);

        //Realizar la resta
        res=n1-n2;
        printf("La resta de %d y %d es: %d \n",n1,n2,res);

    }
    else if(op==3)
    {
        //Solicitar variables
        printf("Dame 2 n%cmeros separados por coma: ",au);
        scanf("%i,%i",&n1,&n2);

        //Realizar la multiplicacion
```

```

res=n1*n2;
printf("La multiplicacion de %d y %d es: %d \n",n1,n2,res);

}
else if(op==4)
{
//Solicitar variables
printf("Dame 2 n%cmeros separados por coma: ",au);
scanf("%i,%i",&n1,&n2);

//Realizar la divicion
res=n1/n2;
printf("La divicion de %d y %d es: %d \n",n1,n2,res);

}
else if(op==5)
{
//Solicitar variables
printf("Dame 2 n%cmeros separados por coma: ",au);
scanf("%i,%i",&n1,&n2);

//Realizar el modulo
res=n1%n2;
printf("El modulo de %d y %d es: %d \n",n1,n2,res);

}
else if(op==6)
{
//Solicitar variables
printf("Ingrese un n%cmero: ",au);
scanf("%d",&n);

//Realizar el facroeial
res=0;
for (int i=n;i>1;i++)
{
    res=res*i;
}
printf("El factorial del n%cmero solicitado es: %d \n",n,au,res);

}
else if(op==7)
{
//Solicitar variables
printf("%cCu%cntos n%cmeros deseas sumar? ",sp,ad,au);
scanf("%d",&n);

//Realizar la sumatoria gauss
res=0;
for(int i=1;i<=n;i++)
{
    res=res+i;
}

printf("La suma de los primeros %d n%cmeros es: %d \n",n,au,res);

```

```
    }  
    else if(op==8)  
    {  
        printf("Elegiste Salir\n\n");  
    }  
  
    return 0;  
}
```


Conclusion

Las estructuras de repetición, como su nombre lo dice nos ayudan a repetir un proceso las veces que lo requiéranos, estas estructuras son fáciles de implementar y muy importantes para programar.

Bibliografía

Solano Gálvez, García Cano, Sandoval Montaña, Nakayama Cervantes, Arteaga Ricci, Castañeda Perdomo, I., 2020. Laboratorio Salas A Y B. [online] Lcp02.fi-b.unam.mx. Available at: <<http://lcp02.fi-b.unam.mx/>> [Accessed 6 April 2020].