```
#include <stdio.h>
     #include <string.h>
3
     // Programación I - Ciclos.
5
      int Ciclos()
6
8
9
         char prof[20];
10
         strcpy(prof, "Miguel Silva C.");
11
12
         printf("Prof. %s", prof);
13
14
15
16
17
         return 0;
18
19
20
21
22
23
```

# // Programación I - Ciclos.

# Concepto Ciclos() {

estructuras de control que permiten ejecutar un bloque de código repetidamente mientras se cumpla una condición específica, o un número determinado de veces. Son fundamentales para la implementación de la repetición y la iteración en un programa.





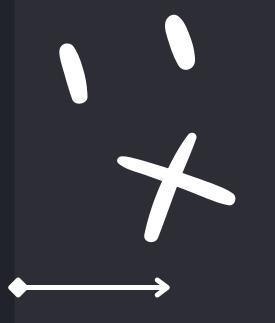
```
// Programación I - Ciclos
2
3
   En C, los ciclos más comunes son:
5
    while (condición)
        While: El bucle while ejecuta
9
        un bloque de código mientras
10
11
               cumpla una condición
        se
12
13
        específica.
14
15
        La condición se evalúa antes de cada
16
17
        iteración.
18
19
20
21
```

```
// Programación I - Ciclos
2
3
   En C, los ciclos más comunes son:
5
    do
      do-while: Similar a while, pero
8
9
10
       la condición se evalúa después de
11
12
       cada iteración.
13
14
      Esto garantiza que el código se ejecute
15
      al menos una vez, incluso si la condición
16
17
       es falsa desde el principio.
18
19
20
    }While(condición);
21
```

```
// Programación I - Ciclos
2
3
   En C, los ciclos más comunes son:
5
    for (inicialización; condición; actualización)
        For: Especifica un contador de
9
        iteraciones, una condición de
10
11
        fin
                    una expresión
                                             de
12
13
        actualización en
                                          sola
                                  una
14
15
        línea.
16
17
        <u>Útil</u> cuando se sabe cuántas veces se
18
        debe repetir el bloque de código.
19
20
    };
21
22
```

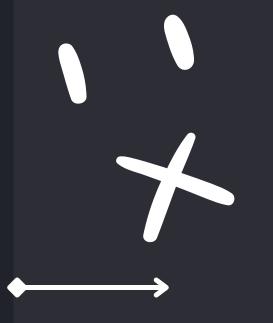
```
// Programación I - Ciclos
 Ejemplo while:
           #include <stdio.h>
                                                Código
 5
              int main()
 6
           4 - {
                  int valor, contador;
 8
           6
 9
                  printf("Ingrese el valor para contar: ");
10
                  scanf("%d", &valor);
           8
11
12
          10
                  contador = 1; //Se inicializa el contador.
13
          11
                  while(contador <= valor)</pre>
          12
14
          13 -
15
          14
                      // Se imprime el contenido del
16
          15
                      // contador en una sola línea.
17
          16
                      printf("%d ", contador);
18
          17
                      // Se incrementa en 1 el contador;
19
          18
                      contador++;
20
          19
21
          20
          21
                  return 0;
22
          22
23
```

```
Ingrese el valor para contar: 10
1 2 3 4 5 6 7 8 9 10
...Program finished with exit code 0
Press ENTER to exit console.
```



```
// Programación T - Ciclos
Ejemplo do-while:
             #include <stdio.h>
                                                Código
 5
            int main()
 6
          4 - {
                 int valor, contador;
 8
          6
 9
                 printf("Ingrese el valor para contar: ");
10
                 scanf("%d", &valor);
          8
11
         10
                 contador = 1; //Se inicializa el contador.
12
         11
13
         12
                 do
14
         13 -
15
                     // Se imprime el contenido del
         14
16
         15
                     // contador en una sola línea.
17
                     printf("%d ", contador);
         16
18
         17
                     // Se incrementa en 1 el contador;
19
         18
                     contador++;
                 }while(contador <= valor);
         19
20
         20
21
         21
                 return 0;
22
         22 }
23
```

```
Ingrese el valor para contar: 10
1 2 3 4 5 6 7 8 9 10
...Program finished with exit code 0
Press ENTER to exit console.
```



```
// Programación I - Ciclos
 Ejemplo for:
      1 #include (stdio.h)
                                             Código
 5
 6
        int main()
     4 - {
            int valor, contador;
 9
10
            printf("Ingrese el valor para contar: ");
11
            scanf("%d", &valor);
12
13
             for(contador=1; contador<=valor; contador++);</pre>
14
                // Se imprime el contenido del
15
                // contador en una sola línea.
16
                printf("%d ", contador);
17
            return 0;
18
    16
19
20
21
22
23
```

```
Ingrese el valor para contar: 10
1 2 3 4 5 6 7 8 9 10
...Program finished with exit code 0
Press ENTER to exit console.
```



```
// Programación T - Ciclos
Ejemplo 2, while:
    #⊥nclude <stdio.h>
                                      Código
6
    int main() {
                                                              Salida
        int contador = 0;
9
                                                   Ejemplo de ciclo while:
10
                                                   Contador: 0
11
        printf("Ejemplo de ciclo while:\n");
                                                   Contador: 1
12
                                                   Contador: 2
13
        while (contador < 5) {</pre>
                                                   Contador: 3
14
            printf("Contador: %d\n", contador);
                                                   Contador: 4
15
16
            contador++;
17
        3
18
19
20
        return O;
21
22
23
```

```
Programación T - ?: los Ejemplo 2, do-while:
      #include <stdio.h>
                                            Código
  5
  6
      int main() {
  8
          int contador = 0;
  9
 10
 11
          printf("Ejemplo de ciclo do-while:\n");
 12
          do [
 13
 14
               printf("Contador: %d\n", contador);
 15
               contador++;
 16
 17
          } while (contador < 5);</pre>
 18
 19
 20
          return 0;
 21
 22
 23
```

```
Ejemplo de ciclo do-while:
Contador: O
Contador: 1
Contador: 2
Contador: 3
Contador: 4
```



```
Programación T - ?: los Ejemplo 2, do-while:
                                            Código
  5
    #include <stdio.h>
    int main() {
                                                                      Salida
        printf("Ejemplo de ciclo for:\n");
                                                         Ejemplo de ciclo for:
 10
        for (int contador = 0; contador < 5; contador++) {</pre>
 11
                                                          Contador: 0
            printf("Contador: %d\n", contador);
 12
                                                         Contador: 1
 13
                                                         Contador: 2
 14
                                                         Contador: 3
 15
        return O;
                                                         Contador: 4
 16
17 }
 18
 19
 20
 21
 22
```

2

4

5

6

# EJERCITACIÓN (Level 1):

- 1. Crear un menú con todos los ejercicios de las prácticas anteriores para que pueda ejecutarse tantas veces como se requiera.
- 2. Crear un programa que calcule la suma de los primeros N números naturales.
- 3. Calcular el factorial de un número ingresado por el usuario.
- 4. Imprimir una tabla de multiplicar para un número ingresado por el usuario (1\*n, 2\*n, 3\*n... n\*n).
- 5. Encontrar el máximo y el mínimo de un conjunto de números ingresados por el usuario.
- 6. Imprimir los números pares del 1 al N (ingresado por el usuario).
- 7. Imprimir los números impares del 1 al N (ingresado por el usuario).
- 8. Ingresar un número entero e indicar cuántos dígitos tiene.



# EJERCITACIÓN (Level 2):

- 1. Calcular e imprimir la suma de los dígitos pares e impares de un número entero ingresado por el usuario.
- 2. Imprimir todos los números primos dentro de un rango dado por el usuario.
- 3. Imprimir los primeros N términos de la serie de Fibonacci.
- 4. Determinar si un número natural ingresado por el usuario es primo.
- 5. Imprimir la secuencia de Collatz para un número ingresado por el usuario.
- 6. Determinar si un número entero ingresado por el usuario es un número de Armstrong (o narcisista).
- 7. Calcular e imprimir la suma de los N primeros términos de la serie armónica.
- 8. Determinar si un número entero ingresado por el usuario es un número perfecto.



# EXPLICACIÓN (Level 2):

#### 1. Fibonacci:

- Explicación: La secuencia de Fibonacci es una serie de números en la que cada número es la suma de los dos números anteriores. Comienza con 0 y 1, y los siguientes números son la suma de los dos números anteriores.
- Fórmula: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...  $F_n$

#### 2. Collatz:

- Explicación: La secuencia de Collatz, también conocida como conjetura de Collatz, es una secuencia generada a partir de un número entero siguiendo ciertas reglas. Si el número es par, se divide por 2; si es impar, se multiplica por 3 y se suma 1.
- Fórmula: No hay una fórmula matemática explícita para la secuencia de Collatz. Se define de manera recursiva como  $a_{n+1} = \begin{cases} a_n/2, & ext{si } a_n ext{ es par} \\ 3a_n+1, & ext{si } a_n ext{ es impar} \end{cases}$ .

### 3. Armstrong o narcisista:

- Explicación: Un número de Armstrong (o narcisista) es un número en el que la suma de las potencias de sus dígitos es igual al propio número.
- Fórmula: Un número de Armstrong de n dígitos,  $abcd\ldots = a^n + b^n + c^n + d^n + \ldots$





# EXPLICACIÓN (Level 2):

### 4. Número armónico:

- Explicación: La serie armónica es una serie infinita en la que cada término es la inversa del número natural correspondiente.
- ullet Fórmula: La serie armónica se representa como  $H_n=1+rac{1}{2}+rac{1}{3}+...+rac{1}{n}$ .

### 5. Número perfecto:

- Explicación: Un número perfecto es un número entero que es igual a la suma de sus divisores propios positivos, excluyendo el número mismo.
- Fórmula: Un número perfecto N es aquel para el cual la suma de sus divisores propios (D) es igual a N, es decir,  $D_1+D_2+\ldots+D_n=N$ , donde D es un divisor propio de N.



### REALIZADO POR MIGUEL SILVA C.



© Esta presentación cuenta con derechos de autor.





