Autoestudio 8: Ciclo For

La estructura **for**, es una estructura de ciclo en donde el control de la repetición está definido precisamente sobre una variable contador.

El ciclo **for** nos permite **encapsular** en una sola instrucción todo el código de la variable contador.

**for (inicialización del contador; condición/expresión ; incremento/decremento del contador)**

instrucción;

**for (inicialización del contador; condición/expresión ; incremento/decremento del contador)**

{

instrucción\_1;

instrucción\_2;

**....**

instrucción\_N;

}

La ejecución de la sentencia **for** sucede así:

1. Se inicializan las variables ***inicialización***.

2. Se evalúa la **condición**.

* 1. Si el resultado de la **condición** es **verdadero**
     1. se ejecutan la(s) instrucción(es)
     2. se ejecuta el **incremento/decremento** y se vuelve al punto.

**2.2** Si el resultado de la condición es **falso**, la ejecución del for se da por finalizada y se continúa con la siguiente instrucción del programa.

Ejemplos:

un ciclo **for** creciente sería:

int contador;

int limite = 5;

**for (contador = 0 contador < limite; contador ++){**

print(“%i”, contador);

}

Un ciclo **for** definiendo un contador más grande y después decrecerlo hasta llegar al límite.

int contador;

int limite = 5;

**for (contador = 30; contador > limite; contador = contador - 5){**

print(“%i”, contador);

}

Sumatoria:

float sumatoria ( int n){ // función que hace la sumatoría hasta n

int k; // k es el contador

float acum = 0;

**for(k =0; k <= n, k++){ // n es el límite**

acum = acum + k + 5; // “acum = acum +” es lo que hace que se vaya acumulando

}

return acum; // devuelve el resultado

}

Multiplicatoria:

float multiplicatoria ( int n){ // función que hace la sumatoría hasta n

int k ; // k es el contador

float acum = 1;

**for(k = 0, k <= n, k++){ // n es el límite**

acum = acum \* ( k + 5 ); // “acum = acum +” es lo que hace que se vaya acumulando

}

return acum; // devuelve el resultado

}

# Si deseas saber más puedes consultar, donde se explican los temas a mayor detalle:

(Joyanes Aguilar, L. y Zahonero Martínez, I. *Programación en C, C++, Java y UML*. México : McGraw Hill, 2010.) las siguientes secciones:

* Repetición: el bucle for. (pp. 160)
* Precauciones en el uso del for. (pp. 165)
* Repetición: el bucle do-while. (pp. 169)
* Comparación de bucles while, for, do-while. (pp. 172)
* Diseño de bucles. (pp. 172)
* Bucles anidados. (pp. 176)