Autoestudio: Ciclo While

# Definición de un Ciclo

Un ciclo es la ejecución repetida de una o varias instrucciones en C se cuenta con 3 tipos de ciclos:

* **while**
* **for**
* **do while**

**while** es una instrucción que ejecuta una sentencia, simple o compuesta, **cero o más** veces, dependiendo del valor de una expresión

**while** (**expr-test** )

instrucción;

o

**while** (**expr-test** )

{

instrucción\_1;

instrucción\_2;

instrucción\_N;

}

Donde **expr-test** es una expresión que evalúa a verdadero o falso, de hecho es una condición igual a la de un if. Mientras la condición sea verdadera el ciclo se seguirá repitiendo. Una **expresión** es cualquier expresión numérica, relacional o lógica.

Las instrucciones dentro del **while** se ejecutan en forma repetida, en secuencia de arriba a abajo, **mientras** la **expresión** sea **verdadera** (recuerda que para c **0** es falso y cualquier otro número es verdadero). Si la **expresión** se evalúa como **falsa** en la primera ocasión, las instrucciones dentro del while ***nunca*** se ejecutan. Si la **expresión** se evalúa como **verdadera** siempre entonces el ciclo es ***eterno*** y el resto del programa ya no se ejecutará.

Ejemplo de ciclo que nunca se ejecuta:

while(0)

printf(“no me ejecuto”);

Ejemplo de ciclo que siempre se ejecuta:

while(1)

printf(“soy eterno”);

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

* 1. Se evalúa la **expresión**.
  2. Si el resultado de la **expresión** es **falso**, la(s) instrucción(es) no se ejecuta y se pasa a ejecutar la siguiente instrucción en el programa.
  3. Si el resultado de la **expresión** es **verdadero**, se ejecuta la(s) instrucción(es) y el proceso se repite comenzando en el punto 1.

**Ver simulación en presentación power point en el archivo anexo!**

## Controlar repeticiones

### Contadores

Hay varias estrategias para controlar el número de repeticiones en un ciclo, una de las más comunes es el uso de una variable de nombre **contador** y un **límite**. La idea general es que el ciclo se ejecute mientras el contador no llegue al límite. Se puede hacer de dos formas incrementando el contador si su valor inicial es menor que el límite,

int contador = 0;

int limite = 5;

while (contador < limite){

print(“%i”, contador);

contador = contador +1;

}

ó definiendo un contador más grande y después decrecerlo hasta llegar al límite.

int contador = 30;

int limite = 5;

while (contador > limite){

print(“%i”, contador);

contador = contador - 5;

}

Observa que en los dos casos el que cambia de valor es el contador y no el límite, esto se debe a que si movemos el valor del límite podemos caer en ciclos eternos.

Adicionalmente es común el uso de una tercera variable llamada **acumulador**. El acumulador lo utilizamos para acumular un valor diferente al contador o al límite, la mayoría de los problemas con ciclos involucran el uso de acumuladores. Por ejemplo el acumulador se usa para calcular series matemáticas como sumatorias y multiplicatorias:

Sumatoria:

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

int k = 0; // k es el contador

**float acum** = 0;

while(k <= n){ // n es el límite

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

k++; // es lo mismo que k = k +1;

}

return acum; // devuelve el resultado

}

Multiplicatoria:

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

int k = 0; // k es el contador

**float acum** = 1;

while(k <= n){ // n es el límite

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

k++; // es lo mismo que k = k +1;

}

return acum; // devuelve el resultado

}

Es importante observar que cuando es una **sumatoria** el acumulador **empieza en 0** y cuando es una **multiplicatoria** el acumulador **empieza en 1**.

### Banderas

Las banderas son variables que cambian su estado si sucede algún evento. La idea es que el ciclo while siempre corra hasta que un evento ocurra, un clásico de este sería un menú que siempre se muestra a menos que el usuario presione S:

int bandera = 0;

char opcion;

int cont = 1;

while(bandera ! = 1){ // la bandera normalmente toma el valor de 1 o 0

print\_menu();

printf(“usted ha visto el menú lleva %i veces, para ordenar presioné (S)”, cont);

scanf(“%c%\*c”, &opcion);

cont++; // contador que lleva la cuante de cuantas repeticiones van

if(opcion == ‘s’ || opcion == ‘S’){ // si el usuario presiona s o S cambia el valor de la bandera

bandera =1;

}

}

# 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:

* Sentencia while. (pp. 151)