Ciclo inexacto

El ciclo while

La finalidad de la palabra reservada while es ejecutar una o varias instrucciones una y otra vez mientras que una condición dada sea cierta. Cuando la condición del ciclo while ya no es lógicamente verdadera, el ciclo termina y la ejecución del programa continúa en la siguiente instrucción después del ciclo.
La forma general de la instrucción while es :

```
while(condición){
  instrucción_1;
  instrucción_2;
          .
          .
  instrucción_N;
}
```

En este caso, condición es la condición de la instrucción while. Esta condición se evalúa primero. Si se evalúa como un valor verdadero entonces se ejecutan las instrucciones. Después de eso, se evalúa nuevamente la condición. Entonces, si la condición se sigue evaluando como verdadera, las instrucciones se ejecutan una vez más. Este proceso se repite una y otra vez hasta que la condición sea evaluada como falsa.

Ejemplo:

```
#include <iostream>
using namespace std;
int main(){
   int n, esPar=true;
   while(esPar==true){
      cout << "Ingresar un numero: ";
      cin >> n;
      if((n%2)==0){ //Acá se determina si el número es divisible por dos.
         esPar = true;
         cout << "El número " << n << " es par";
      }
      else{
         esPar = false;
      }
      cout << endl << endl;
   }
}
```

T05CF01.cpp

Salida:

```
Ingresar un número: 10 ↵
El número 10 es par
Ingresar un número: 4 ↵
El número 4 es par
Ingresar un número: 842 ↵
El número 842 es par
```

```
Ingresar un número: 5 ↵
```

En el ejemplo anterior, el ciclo while se encarga de solicitar números y evaluarlos. La idea es que el programa pida el ingreso de números hasta que el usuario ingrese un número impar.
Para ello se utilizan dos variables:
n – para solicitar los números.
esPar – para controlar el ciclo while, cuando esPar contenga false o cero, entonces el while finalizará.

Se puede observar que en la línea 4 se declara la variable esPar como entero y se la inicializa con el valor true. Esto se hace para "obligar" a entrar al while ya que la condición siempre será evaluada por primera vez como verdadero.

```
true == true → true
```

Luego se ingresa en la variable n el valor del número a ser evaluado y se evalúa si el resto de la división entre el número ingresado y dos es cero, si es cero significa que el número es par, por lo tanto, asignamos a la variable esPar el valor verdadero y se emite un mensaje por pantalla indicando el número ingresado.
En caso que el número no sea par (el resultado del resto es uno), asignamos a la variable esPar el valor falso lo que finalizará el ciclo while.

```
false == true → false
```

El ciclo do-while

En la instrucción while que acabamos de ver, la expresión condicional se coloca al principio del ciclo. Sin embargo, en esta otra instrucción do-while la expresión condicional se pone al final del ciclo. De esta forma, se garantiza que las instrucciones del ciclo se ejecuten por lo menos una vez antes de verificar la condición.

La forma general del ciclo do-while es la siguiente:

```
do{
  instrucción_1;
  instrucción_2;
          .
          .
  instrucción_N;
} while(condición);
```

Aquí, las instrucciones que están dentro del bloque se ejecutan una vez, y luego se evalúa la condición a fin de determinar si el ciclo continúa. El ciclo do-while continúa si la expresión se evalúa como un valor verdadero, en caso contrario, la ejecución sigue en la instrucción después del ciclo.

Nota: La instrucción do-while termina con un punto y coma, lo cual es una diferencia importante con respecto a las instrucciones if y while.

Ejemplo:

```
#include <iostream>
using namespace std;
int main(){
```

```
   int n, esPar=false;
   do{
      cout << "Ingresar un numero: ";
      cin >> n;
      if((n%2)==0){ //Acá se determina si el número es divisible por dos.
         esPar = true;
         cout << "El número " << n << " es par";
      }
      else{
         esPar = false;
      }
      cout << endl << endl;
   }while(esPar==true);
}
```

T05CF02.cpp

Salida:

```
Ingresar un número: 10 ↵
El número 10 es par
Ingresar un número: 4 ↵
El número 4 es par
Ingresar un número: 842 ↵
El número 842 es par
Ingresar un número: 5 ↵
```

Este ejemplo funciona de la misma manera que el ejemplo T05CF01.cpp, sólo que aquí se utiliza el ciclo do-while. Se puede notar la diferencia en que la variable esPar que en el ejemplo primero se inicializaba como verdadero para "obligar" al ciclo a evaluarla como verdadero, aquí se inicializa como falso (puede contener cualquier valor ya que es indiferente) para demostrar que sin importar el valor de dicha variable, el conjunto de instrucciones del ciclo do-while se ejecutará al menos una vez.

Bibliografía:

- Zhang, Tony, Aprendiendo C en 24 horas, Pearson Educación, México, 2001, ISBN: 968-444-495-8.