

Electrónica Digital II

Santiago Rúa Pérez, PhD.

18 de septiembre de 2022

PROGRAMACIÓN ESTRUCTURADA EN C

Programación estructurada en C

Objetivos

- Entender los bucles por centinela y contador.
- Entender ciclos for.
- Entender ciclos do while.
- Comando break and continue.
- Operadores lógicos.

Iteracion controlada por contador

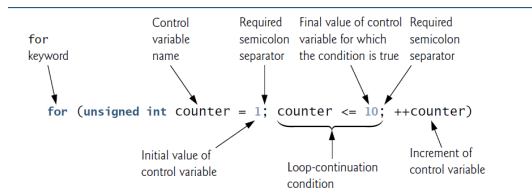
- El nombre de una variable de control.
- Un valor inicial de esa variable de control.
- Un incremento o decremento de la señal de control.
- Una condicion de valor final.

```
1 // Iteracion controlada por contador
2 #include <stdio.h>
3
4 int main(void)
5 {
6     unsigned int counter = 1;
7
8     while(counter <= 10){
9         printf("%u\n", counter);
10        ++counter;
11    }
12 } // Fin de la funcion
13
```

En las iteraciones controladas por controlador, se sabe cuantas veces se va a ejecutar el *loop*. En las controladas por centinela no, ya que se espera que se cumpla una condición particular y no se puede predecir cuantas veces se ejecutara el *loop*.

Ciclos for

- Contiene tres partes: condicion inicial, condicion de paro, y condicion de incremento



Ejemplo: Hacer un ciclo for que haga lo siguiente:

- Que varíe la señal de control en pasos de 7, y que vaya desde 7 hasta 77
- Que vaya de 20 a 2, en decremento de a 2.
- Varíe la señal de control sobre la siguiente secuencia 44,33,22,11,0.

Ciclos for - Ejemplo

Haga un programa que calcule el interes compuesto dado un valor de inversión. El interes compuesto esta dado por $a = p(1 + r)^n$, donde p es el valor original invertido, r es la tasa de interes anual, n el número de años, a es el deposito al final del año n .

Solución - ejemplo

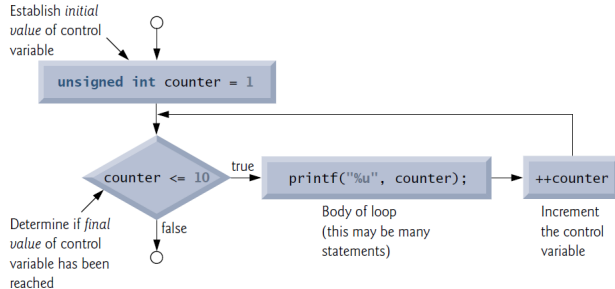
```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void)
5  {
6      double principal = 1000.0; // starting principal
7      double rate = .05; // annual interest rate
8
9      // output table column heads
10     printf("% 4s% 21s\n", "Year", "Amount on deposit");
11
12     // calculate amount on deposit for each of ten years
13     for (unsigned int year = 1; year <= 10; ++year) {
14
15         // calculate new amount for specified year
16         double amount = principal * pow(1.0 + rate, year);
17
18         // output one table row
19         printf("% 4u% 21.2f\n", year, amount);
20     }
21 }
22
```

Observaciones ciclos for

- La inicialización, la condición de iteración y el incremento pueden contener operaciones aritméticas. Por ejemplo:

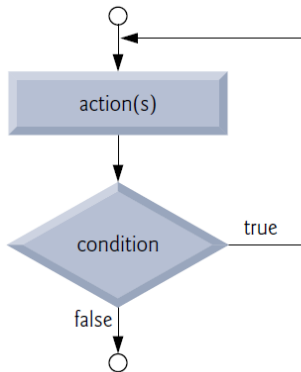
```
1  for (j = x; j <= 4 * x * y; j += y / x)
```

- El incremento puede ser negativo. Cuenta de forma descendente.
- Si la condición del *loop* es falsa al inicio, el for no se ejecuta ninguna vez.
- El usual no usar la variable de control dentro del loop.
- El ciclo *for* es como una secuencia de flujo.



Ciclos do... while

Es muy similar al ciclo while, con la variante que se asegura ejecutar al menos una vez el ciclo.



En el diagrama se observa que la acción dentro del ciclo se ejecuta y luego se realiza la pregunta de la condición.

Ejemplo do... while

```
1  #include <stdio.h>
2
3  int main( void )
4  {
5      unsigned int counter = 1;
6
7      do{
8          printf("%u ", counter);
9      }while(++counter<=10);
10 }
11
```

En el ejemplo se tiene un ciclo controlado por contador

Nota: Las palabras **break** y **continue** son utilizadas en C para terminar un ciclo de forma abrupta o saltarse una iteración del mismo.

Ejemplo break

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      unsigned int x; // declared here so it can be used after loop
6
7      // loop 10 times
8      for (x = 1; x <= 10; ++x) {
9          // if x is 5, terminate loop
10         if (x == 5) {
11             break; // break loop only if x is 5
12         }
13         printf("%u ", x);
14     }
15     printf("\nBroke out of loop at x == %u\n", x);
16 }
17
```

A pesar de que el *for* esta estructurado para ejecutarse 10 veces, no lo hará debido a la instrucción *break* con la condición que se impone.

Ejemplo continue

```
1 // Using the continue statement in a for statement.
2 #include <stdio.h>
3
4 int main(void)
5 {
6     // loop 10 times
7     for (unsigned int x = 1; x <= 10; ++x) {
8         // if x is 5, continue with next iteration of loop
9         if (x == 5) {
10             continue; // skip remaining code in loop body
11         }
12         printf("%u ", x);
13     }
14     puts("\nUsed continue to skip printing the value 5");
15 }
16
```

Operadores Lógicos

Operador lógico AND

```
1  if (gender == 1 && age >= 65) {  
2      ++seniorFemales;  
3  }  
4
```

Operador lógico OR

```
1  if (semesterAverage >= 90 || finalExam >= 90) {  
2      puts("Student grade is A");  
3  }  
4
```

Operador lógico negacion

```
1  if (!(grade == sentinelValue)) {  
2      printf("The next grade is %f\n", grade);  
3  }  
4
```

Detección de errores

- a) `x = 1;`
`while (x <= 10);`
`++x;`
`}`
- b) `for (double y = .1; y != 1.0; y += .1) {`
`printf("%f\n", y);`
`}`
- c) `switch (n) {`
`case 1:`
`puts("The number is 1");`
`case 2:`
`puts("The number is 2");`
`break;`
`default:`
`puts("The number is not 1 or 2");`
`break;`
`}`
- d) The following code should print the values 1 to 10.

```
n = 1;
while (n < 10) {
    printf("%d ", n++);
}
```

Ejercicios

- Suma y promedio de enteros: escriba un programa que sume la secuencia de número enteros y calcule el promedio. Asuma que el primer entero indicará cuantos enteros va a leer. Tu programa debe leer un numero al tiempo.
- Haga un programa que convierta de Farenheit a Celsius.
- Haga un programa que calcule el factorial de un número.
- Haga un programa que obtenga todos los números primos del 1 al 100.
- Escriba un programa que imprima la suma, la suma del cuadrado y la suma del cubo de todos los números naturales del 1 hasta el numero ingresado por el usuario.

PROGRAMACIÓN EN C

GRACIAS