

Programación II
Ingeniería en Sistemas de Información Sabatino
Departamento de Computación
Facultad de Ciencias y Tecnología

UNIDAD IV:

ESTRUCTURAS DE CONTROL EN C
(for, while)

UNIDAD IV: ESTRUCTURAS DE CONTROL: BUCLE FOR

4.1 INTRODUCCIÓN

Una de las características de las computadoras que aumentan considerablemente su potencia es su capacidad para ejecutar una tarea muchas veces con gran velocidad, precisión y fiabilidad. Las tareas repetitivas es algo que los humanos encontramos difíciles y tediosas de realizar. C soporta tres tipos de estructuras de control: bucles **for**, **while** y **do-while**.

Un bucle (ciclo) es cualquier construcción de programa que repite una sentencia o secuencia de sentencias un número de veces. La sentencia (o grupo de sentencias) que se repiten en un bloque se denomina cuerpo del bucle y cada repetición del cuerpo se llama iteración del bucle. Las dos principales cuestiones de diseño en la construcción del bucle son: **¿Cuál es el cuerpo del bucle?** **¿Cuántas veces se iterará el cuerpo del bucle?**

Existen tres tipos de estructuras:

- **while.**
- **do while.**
- **for**

Estas estructuras nos permiten repetir unas instrucciones concretas un número indeterminado (o no) de veces, hasta que se cumpla una condición.

4.2 DECLARACIÓN Y USO

```
int i = 0;
while(i<3)
{
    printf("i vale %d\n",i);
    i++;
}
```

```
int i =0;
do
{
    printf("i vale %d\n",i);
    i++;
}while(i<3);
```

```
int i;
for(i=0;i<3;i++)
{
    printf("i vale %d\n",i);
}
```

Salida:

i vale 0
i vale 1
i vale 2

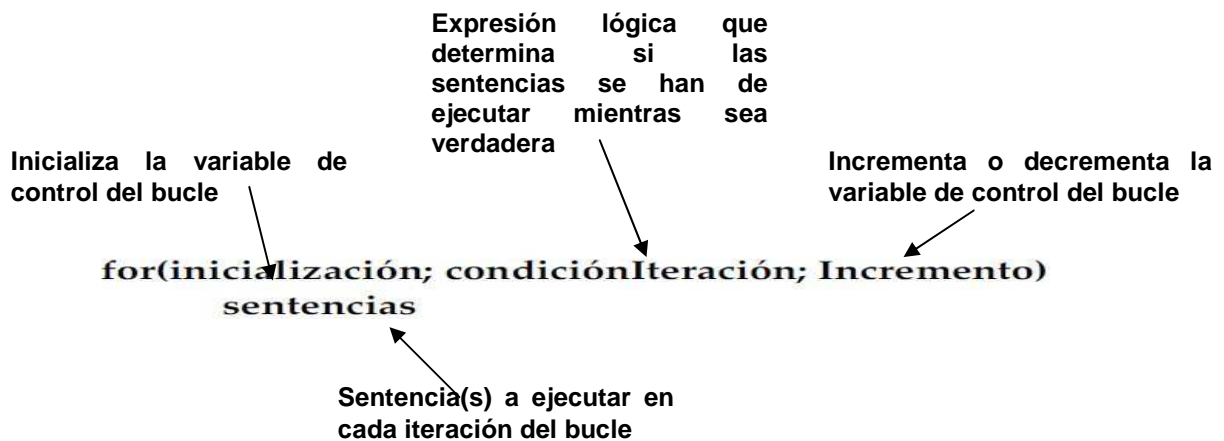
Estos tres ejemplos producen la misma salida, pero su uso general es muy distinto.

4.3 REPETICIÓN: BUCLE FOR

El bucle **for** de C es el más adecuado para implementar bucles controlados por contador que son bucles en los que un conjunto de sentencias se ejecutan una vez por cada valor de un rango especificado, de acuerdo al algoritmo:

Por cada valor de una variable_contador de un rango específico: ejecutar sentencias

La sentencia **for (bucle for)** es un método para ejecutar un bloque de sentencias un número fijo de veces. El bucle **for** se diferencia del bucle **while** en que las operaciones de control del bucle se sitúan en la cabecera de la sentencia.



El bucle **for** contiene las siguientes partes:

- **Parte de inicialización:** Que inicializa la variable de control del bucle.
- **Parte de condición:** Contiene una expresión lógica que hace que el bucle realice las iteraciones de las sentencias, mientras que la expresión sea verdadera.
- **Parte de Incremento o Decremento:** Incrementa o decrementa la(s) variable(s) de control del bucle.
- **Sentencias:** Acciones o sentencias que se ejecutará por cada iteración del bucle.

En el bucle **for** el número de iteraciones se conoce antes de ejecutarse el ciclo. La forma de esta estructura es la siguiente:

Hacer para V.C = V.I a L.S V.C = V.L

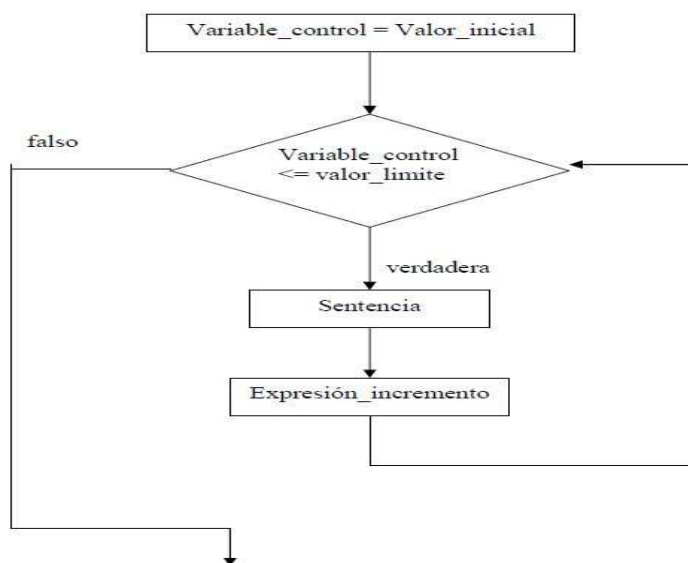
Accion₁
 Accion₂.

 Accion_N

Fin-para

Donde:

- ◆ V.C: Variable de Control
- ◆ V.I: Valor Inicial
- ◆ V.L: Valor Límite



En este ciclo la variable de control toma el valor inicial del ciclo y el ciclo se repite hasta que la variable de control llegue al valor límite.

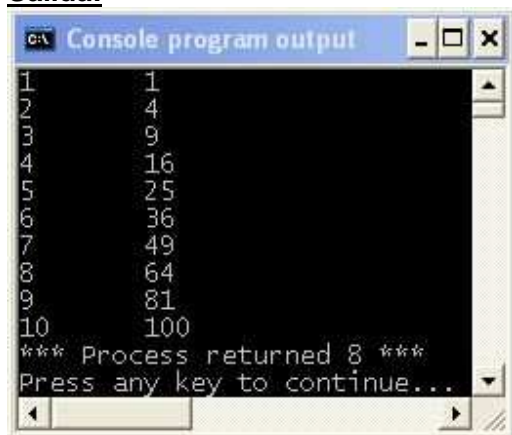
Existen dos formas de implementar la sentencia **for** que se utilizan normalmente para implementar bucles de conteo: **formato ascendente** en el que la variable de control se incrementa y **formato descendente** en el que la variable de control se decrementa.

4.4 **Formato Ascendente:**

```
for(var_control = valor_inicial; var_control <= valor_límite; exp_incremento)
    sentencia;
```

```
//forasc.c
#include<stdio.h>
int vc; //Declaración de una variable global
void main()
{
    for(vc=1;vc<=10;vc++)
        printf("%d\t%d\n",vc,vc* vc);
}
```

Salida:



```
c:\ Console program output
1      1
2      4
3      9
4     16
5     25
6     36
7     49
8     64
9     81
10    100
*** Process returned 8 ***
Press any key to continue...
```

4.5 **Formato Descendente:**

```
for(var_control = valor_inicial; var_control >= valor_límite; exp_decremento)
    sentencia;
```

```
//fordesc.c
#include<stdio.h>
int vc; //Declaración de una variable global
void main()
{
    for(vc =10; vc >=1; vc --)
        printf("%d\t%d\n", vc, vc * vc);
}
```

Salida:



```
C:\> Console program output
10      100
9       81
8       64
7       49
6       36
5       25
4       16
3       9
2       4
1       1
*** Process returned 5 ***
```

4.6 Otros intervalos de incremento y decremento

Los rangos de incremento/decremento de la variable o expresión de control del bucle pueden ser cualquier valor y no siempre 1, es decir: 2, 3, 5, 7, 20, 30... dependiendo de los intervalos que se necesiten.

```
//forintervaloinc.c
#include<stdio.h>
int n; //Declaración de una variable global
void main()
{
    for(n=0;n<100;n+=20)
        printf("%d\t%d\n",n,n*n);
}
```

Salida:



```
C:\> Console program output
0       0
20      400
40      1600
60      3600
80      6400
*** Process returned 9 ***
Press any key to continue...
```

Ejemplos:

- Este es un programa en C que permite: Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de Programación II.

```
//promedio.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int suma=0,i=1;
    float calif, prom;
    char nombre[30];
    printf("Introduzca el nombre: ");
    scanf("%s",&nombre);
    for(i=1;i<=7;i++)
    {
        printf("Calificacion[%d]: ",i);
        scanf("%f",&calif);
        suma = suma + calif;
    }
    prom = suma /7;
    printf("\n%s Tiene un PROMEDIO de: %.2f\n",nombre,prom);
}
```

Salida:



```
Console program output
Introduzca el nombre: Luis
Calificacion[1]: 45
Calificacion[2]: 67
Calificacion[3]: 89
Calificacion[4]: 78
Calificacion[5]: 56
Calificacion[6]: 43
Calificacion[7]: 23

Luis Tiene un PROMEDIO de: 57.00
```

- Este es un programa en C que permite: Leer N números e imprimir solamente los números positivos.

```
//pos_neg.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int i=1,num,ndatos;
    printf("Cuantos Datos: ");
    scanf("%d", &ndatos);
```

```
for(i=1;i<=ndatos;i++)
{
    printf("Numero[%d]: ",i);
    scanf("%d",&num);
    if(num >0)
        printf("%d ES POSITIVO\n",num);
    else
        printf("%d NO ES POSITIVO\n",num);
}
}
```

Salida:

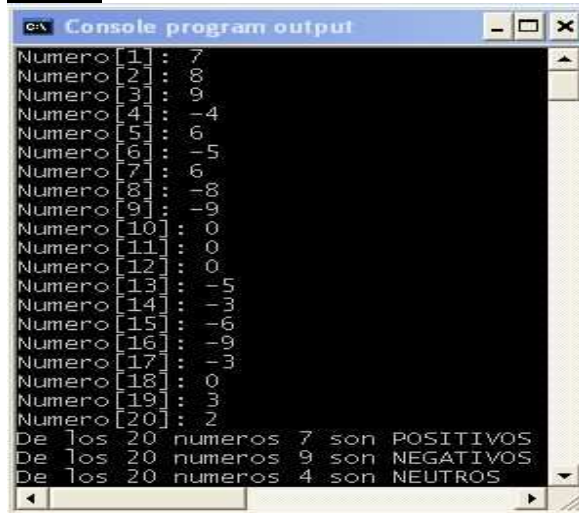


```
Cuantos Datos:7
Numero[1]: 5
5 ES POSITIVO
Numero[2]: -6
-6 NO ES POSITIVO
Numero[3]: -7
-7 NO ES POSITIVO
Numero[4]: 8
8 ES POSITIVO
Numero[5]: 0
0 NO ES POSITIVO
Numero[6]: 3
3 ES POSITIVO
Numero[7]: -5
-5 NO ES POSITIVO
```

- Este es un programa en C que permite leer 20 números e imprimir cuantos son positivos, cuantos negativos y cuantos neutros.

```
//n_pos_neg_neu.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int i=1,num,pos=0,neg=0,neu=0;
    for(i=1;i<=20;i++)
    {
        printf("Numero[%d]: ",i);
        scanf("%d",&num);
        if(num >0)
            pos++;
        else if(num <0)
            neg++;
        else
            neu++;
    }
    printf("De los 20 numeros %d son POSITIVOS\n",pos);
    printf("De los 20 numeros %d son NEGATIVOS\n",neg);
    printf("De los 20 numeros %d son NEUTROS\n",neu);
}
```

Salida:



```
Numero [1] : 7
Numero [2] : 8
Numero [3] : 9
Numero [4] : -4
Numero [5] : 6
Numero [6] : -5
Numero [7] : 6
Numero [8] : -8
Numero [9] : -9
Numero [10] : 0
Numero [11] : 0
Numero [12] : 0
Numero [13] : -5
Numero [14] : -3
Numero [15] : -6
Numero [16] : -9
Numero [17] : -3
Numero [18] : 0
Numero [19] : 3
Numero [20] : 2
De los 20 numeros 7 son POSITIVOS
De los 20 numeros 9 son NEGATIVOS
De los 20 numeros 4 son NEUTROS
```

- Suponga que se tiene un conjunto de calificaciones de un grupo de 5 alumnos. Realizar un programa en C para calcular la calificación promedio y la calificación más baja de todo el grupo.

//calif_menor.c

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

```
void main()
```

```
{
```

```
    int suma = 0, baja=9999, i=1;
```

```
    float calif,media;
```

```
    for(i=1;i<=5;i++)
```

```
    {
```

```
        printf("Calificacion[%d]: ",i);
```

```
        scanf("%f",&calif);
```

```
        suma = suma + calif;
```

```
        //Encontrar cual es la calificacion menor
```

```
        if(calif< baja)
```

```
            baja = calif;
```

```
    }
```

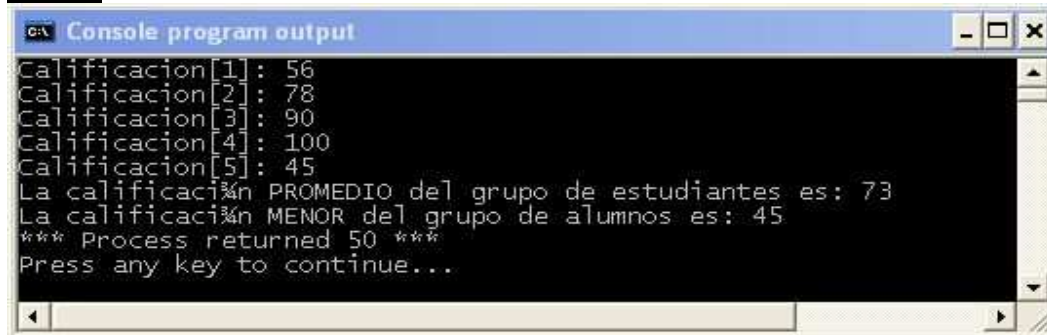
```
    media = suma/5;
```

```
    printf("La calificación PROMEDIO del grupo de estudiantes es: %.f\n",media);
```

```
    printf("La calificación MENOR del grupo de alumnos es: %d\n",baja);
```

```
}
```

Salida:



```
C:\ Console program output
Calificacion[1]: 56
Calificacion[2]: 78
Calificacion[3]: 90
Calificacion[4]: 100
Calificacion[5]: 45
La calificación PROMEDIO del grupo de estudiantes es: 73
La calificación MENOR del grupo de alumnos es: 45
*** Process returned 50 ***
Press any key to continue...
```

- Este es un programa en C que permite: Calcular e imprimir la tabla de multiplicar de un número cualquiera. Imprimir el multiplicando, el multiplicador y el producto.

```
//tabla_mul.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int i=1, num;

    printf("DESEO MOSTRAR LA TABLA DE MULTIPLICAR DEL: ");
    scanf("%d",&num);
    //system("cls");
    printf("\n*****TABLA DE MULTIPLICAR DEL: %d*****\n",num);
    for(i=1;i<=10;i++)
        printf("%d * %d = %d\n",num,i,num*i);
}
```

Salida:



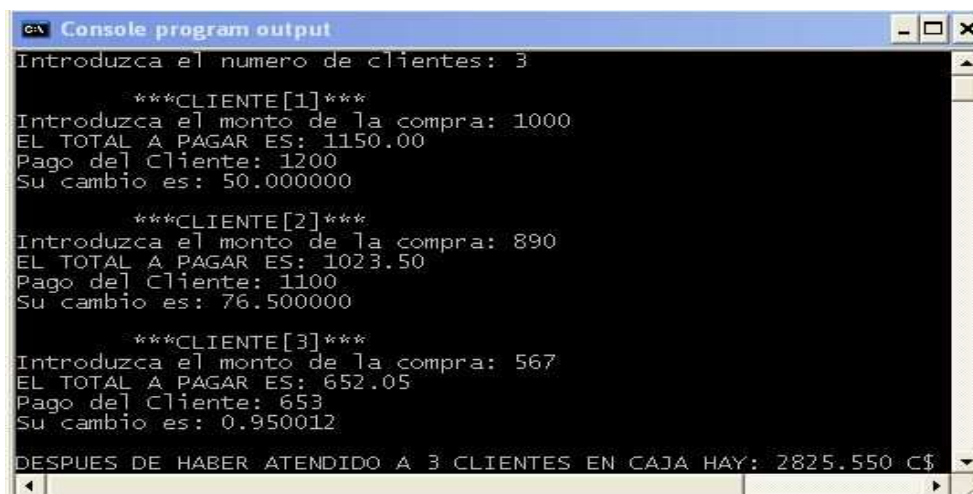
```
C:\ Console program output
DESEO MOSTRAR LA TABLA DE MULTIPLICAR DEL: 6
*****TABLA DE MULTIPLICAR DEL: 6*****
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
6 * 10 = 60
```

4.7 PROBLEMAS PROPUESTOS DE LA ESTRUCTURA DE CONTROL: for

1. Realice un programa en C que imprima los números pares e impares entre 1 y 100.
2. Realice un programa que permita calcular el factorial de un número.
3. Realice un programa en C que calcule la suma de los números impares desde 1 hasta un número dado N.
4. Se desea realizar una estadística de los pesos de los 150 alumnos de un colegio de acuerdo a la siguiente tabla:
 - a. Alumnos de menos de 40 kg.
 - b. Alumnos entre 40 y 50 50kg.
 - c. Alumnos mayores de 50 kg y menores de 60 kg.
 - d. Alumnos mayores de 60 kg.
5. Realice un programa en C para calcular la suma siguiente: $100 + 98 + 96 + 94 + \dots + 0$ en este orden.
6. Desarrollar un programa en C que determine en un conjunto de 100 números naturales ¿cuántos son menores de 10, mayores de 50 y cuántos están comprendidos entre 100 y 200?
7. Suponga que tiene una tienda y desea registrar sus ventas por medio de una computadora. Diseñe un programa en C que permita realizar las siguientes operaciones para N clientes:
 - a) El monto de la compra de cada cliente.
 - b) Calcule e imprima el IVA (15% de la compra).
 - c) Calcule e imprima el total a pagar.
 - d) Lea la cantidad con que paga el cliente.
 - e) Calcule e imprime el cambio.

Al final del día deberá imprimir la cantidad de dinero que debe haber en la caja.

Ejemplo de Entrada y salida de datos:



```
ca Console program output
Introduzca el numero de clientes: 3
***CLIENTE[1]***
Introduzca el monto de la compra: 1000
EL TOTAL A PAGAR ES: 1150.00
Pago del Cliente: 1200
Su cambio es: 50.000000
***CLIENTE[2]***
Introduzca el monto de la compra: 890
EL TOTAL A PAGAR ES: 1023.50
Pago del Cliente: 1100
Su cambio es: 76.500000
***CLIENTE[3]***
Introduzca el monto de la compra: 567
EL TOTAL A PAGAR ES: 652.05
Pago del Cliente: 653
Su cambio es: 0.950012
DESPUES DE HABER ATENDIDO A 3 CLIENTES EN CAJA HAY: 2825.550 C$
```

4.8 SENTENCIA DE CONTROL: while

Un bucle **while** tiene una condición del bucle (una expresión lógica) que controla la secuencia de repetición. La posición de esta condición del bucle es delante del cuerpo del bucle, de modo que cuando se ejecuta el mismo, se evalúa la condición antes de que se ejecute el cuerpo del bucle.

El diagrama indica que la ejecución de la sentencia o sentencias expresadas se repite mientras la condición del bucle permanece verdadera y termina cuando se hace falsa. Si la condición es inicialmente falsa, el cuerpo del bucle no se ejecutará. En otras palabras, el cuerpo de un bucle **while** se ejecutará cero o más veces.

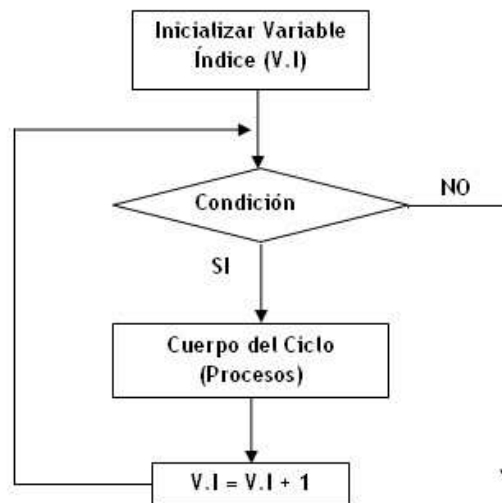
La forma de esta estructura es la siguiente:

```

Hacer mientras <condición>
    Accion1
    Accion2
    .
    .
    AccionN
Fin-mientras

while(condición_bucle)
    sentencia;           → cuerpo

while(condición_bucle)
{
    sentencia-1;
    sentencia-2;
    .
    .
    sentencia-n;
}
    
```



Donde:

condición_bucle es una expresión lógica o booleana
sentencia es una sentencia simple o compuesta

El funcionamiento de una sentencia **while** es:

- Se evalúa la condición_bucle
- Si condición_bucle es verdadera (distinto de cero):
 - a) La sentencia especificada, denominada el cuerpo del bucle, se ejecuta.
 - b) Vuelve el control al paso 1.
- En caso contrario: El control se transfiere a la sentencia siguiente al bucle **while**.

Las sentencias del cuerpo del bucle se repiten mientras que la expresión lógica (condición del bucle) sea verdadera. Cuando se evalúa la expresión lógica y resulta falsa, se termina y se sale del bucle y se ejecuta la siguiente sentencia de programa después de la sentencia **while**.

4.9 **BUCLAS while CON CERO ITERACIONES**

El cuerpo de un bucle no se ejecuta nunca si la condición de repetición del bucle no se cumple, es falsa(es cero en C), cuando se alcanza while la primera vez.

```
contador = 10;
while(contador > 100)
{
    ...
}
```

El bucle anterior nunca se ejecutará ya que la condición del bucle (contador > 100) es falsa la primera vez que se ejecuta. El cuerpo del bucle nunca se ejecutará.

4.10 **BUCLAS CONTROLADOS POR CENTINELAS**

Normalmente, no se conoce con exactitud cuántos elementos de datos se procesarán antes de comenzar su ejecución. Esto se produce bien porque hay muchos datos a contar normalmente o porque el número de datos a procesar depende de cómo prosigue el proceso de cálculo.

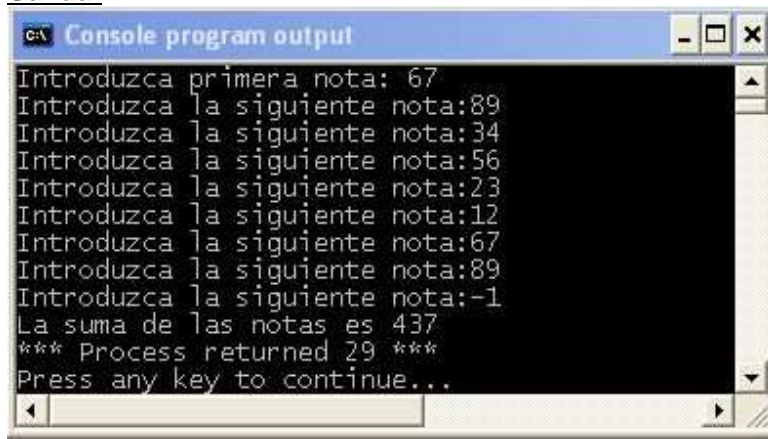
Un medio para manejar esta situación es instruir al usuario a introducir un único dato definido y especificado denominado valor centinela como último dato. La condición del bucle comprueba cada dato y termina cuando se lee el valor centinela. En realidad el centinela es un valor que sirve para terminar el proceso del bucle.

En el siguiente fragmento de código hay un bucle con centinela; se introducen notas mientras que ésta sea distinta de centinela.

```
//centinela.c
/*entrada de datos numéricos, centinela -1*/
#include <stdio.h>
void main()
{
    int nota,suma = 0;
    const int centinela = -1;
    printf("Introduzca primera nota: ");
    scanf("%d",&nota);
    while(nota != centinela)
    {
        suma += nota;
        printf("Introduzca la siguiente nota:");
        scanf("%d",&nota);
    }/* fin de while*/

    puts("Final");
    printf("La suma de las notas es %d \n",suma);
}
```

Salida:



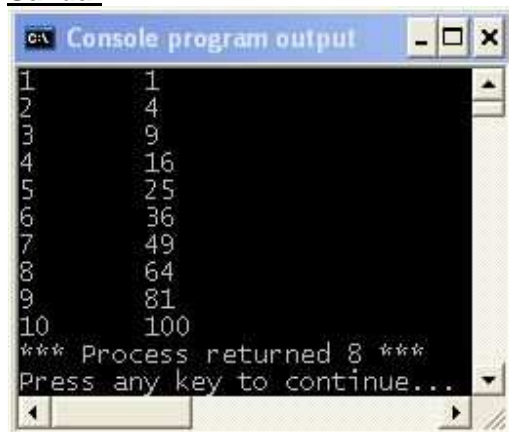
```
c:\ Console program output
Introduzca primera nota: 67
Introduzca la siguiente nota:89
Introduzca la siguiente nota:34
Introduzca la siguiente nota:56
Introduzca la siguiente nota:23
Introduzca la siguiente nota:12
Introduzca la siguiente nota:67
Introduzca la siguiente nota:89
Introduzca la siguiente nota:-1
La suma de las notas es 437
*** Process returned 29 ***
Press any key to continue...
```

4.11 PROBLEMAS RESUELTOS CON LA ESTRUCTURA DE CONTROL: while

1. Programa que imprime los números del 1 al 10 en formato ascendente.

```
//whileasc.c
#include<stdio.h>
void main()
{
    int vc=1;
    while(vc <=10)
    {
        printf("%d\t%d\n", vc, vc * vc);
        vc++;
    }
}
```

Salida:

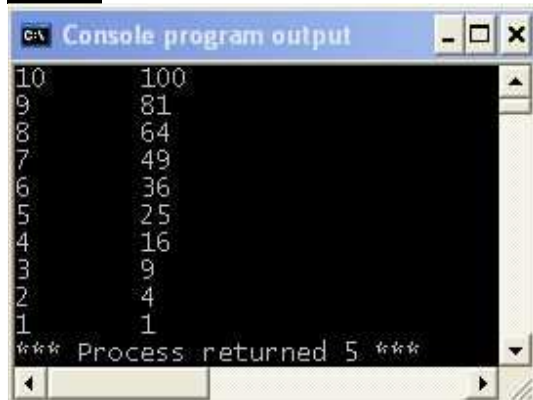


```
c:\ Console program output
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
*** Process returned 8 ***
Press any key to continue...
```

2. Programa que imprime los números del 1 al 10 en formato descendente.

```
//whiledesc.c
#include<stdio.h>
void main()
{
    int vc=10;
    while(vc >=1)
    {
        printf("%d\t%d\n", vc, vc * vc);
        vc--;
    }
}
```

Salida:




```
10      100
9       81
8       64
7       49
6       36
5       25
4       16
3       9
2       4
1       1
*** Process returned 5 ***
```

3. Realice un programa en C que permita: Obtener el promedio de calificaciones de un grupo de **N** alumnos.

```
//promedio.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int suma=0,a=1,n_alumnos;
    float calif, prom;
    printf("Introduzca el numero de alumnos: ");
    scanf("%d",&n_alumnos);
    while(a<=n_alumnos)
    {
        printf("Calificacion [%d]: ",a);
        scanf("%f",&calif);
        suma = suma + calif;
        a++;
    }
    prom = suma /n_alumnos;
    printf("\nEL PROMEDIO DE NOTAS ES:%.2f",prom);
}
```

Salida:

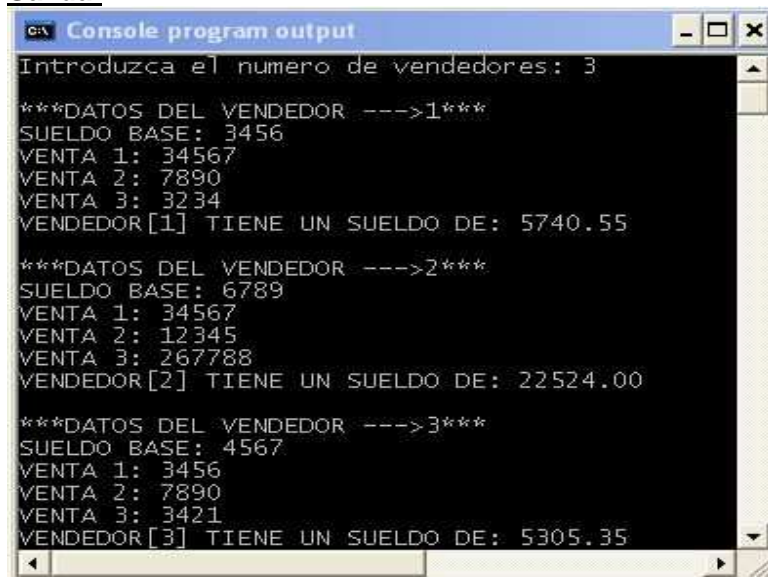


The screenshot shows a console window titled "Console program output". The text inside the window is as follows:
Introduzca el numero de alumnos: 4
Calificacion [1]: 45
Calificacion [2]: 67
Calificacion [3]: 89
Calificacion [4]: 78
EL PROMEDIO DE NOTAS ES:69.00*** Process returned 30 ***

4. Una compañía de seguros tiene contratados a **N** vendedores. Cada uno hace tres ventas a la semana. Su política de pagos es que un vendedor recibe un sueldo base, y un 5% extra por comisiones de sus ventas. El gerente de su compañía desea saber cuanto dinero obtendrá en la semana cada vendedor por concepto de comisiones por las tres ventas realizadas, y cuanto tomando en cuenta su sueldo base y sus comisiones.

```
//vendedores.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int v=1,n_vendedores;
    float sueldo_base,v1,v2,v3,comisiones,total_sueldo;
    printf("Introduzca el numero de vendedores: ");
    scanf("%d",&n_vendedores);
    while(v<=n_vendedores)
    {
        printf("\n***DATOS DEL VENDEDOR --->%d***\n",v);
        printf("SUELDO BASE: ");
        scanf("%f",&sueldo_base);
        printf("VENTA 1: ");
        scanf("%f",&v1);
        printf("VENTA 2: ");
        scanf("%f",&v2);
        printf("VENTA 3: ");
        scanf("%f",&v3);
        comisiones = (v1+ v2+ v3) * 0.05;
        total_sueldo = sueldo_base + comisiones;
        printf("VENDEDOR[%d] TIENE UN SUELDO DE: %.2f\n",v,total_sueldo);
        v++;
    }
}
```

Salida:



```
Console program output
Introduzca el numero de vendedores: 3

***DATOS DEL VENDEDOR --->1***
SUELDO BASE: 3456
VENTA 1: 34567
VENTA 2: 7890
VENTA 3: 3234
VENDEDOR [1] TIENE UN SUELDO DE: 5740.55

***DATOS DEL VENDEDOR --->2***
SUELDO BASE: 6789
VENTA 1: 34567
VENTA 2: 12345
VENTA 3: 267788
VENDEDOR [2] TIENE UN SUELDO DE: 22524.00

***DATOS DEL VENDEDOR --->3***
SUELDO BASE: 4567
VENTA 1: 3456
VENTA 2: 7890
VENTA 3: 3421
VENDEDOR [3] TIENE UN SUELDO DE: 5305.35
```

5. Encontrar el mayor valor de un conjunto de **N** números dados.

```
//mayor.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int d=1, mayor=0, ndatos,valor;

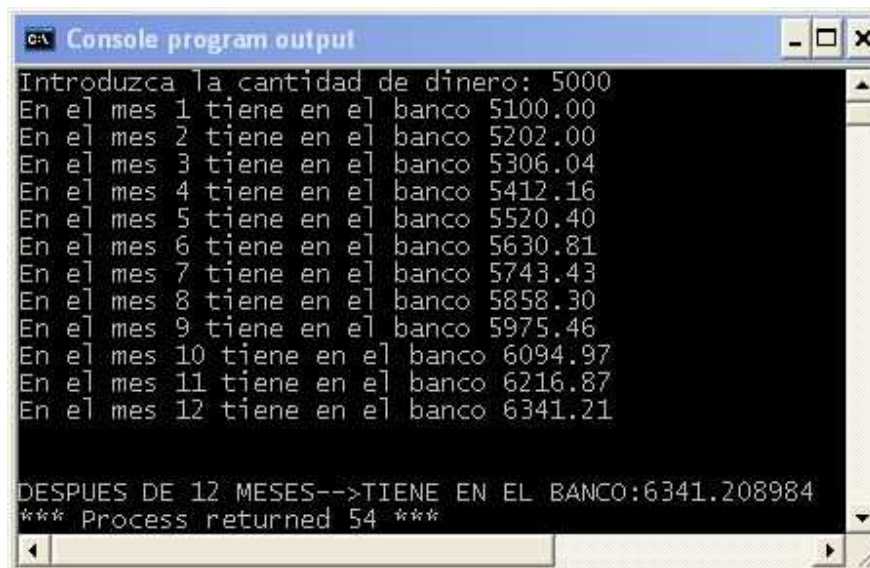
    printf("Introduzca el numero de datos: ");
    scanf("%d",&ndatos);
    while(d<=ndatos)
    {
        printf("Introduzca el valor: ");
        scanf("%d",&valor);
        if(valor > mayor)
            mayor = valor;
        d++;
    }
    printf("EL NUMERO MAYOR ES:%d\n",mayor);
}
```

6. Una persona desea invertir su dinero en un banco, el cual le otorga un 2% de interés. Realice un programa en C que le permita a esa persona saber: Cual será la cantidad de dinero que tendrá al cabo de un año si la ganancia de cada mes es reinvertida?

```
//intereses.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int mes=1;
    float cantdinero, intereses=0;

    printf("Introduzca la cantidad de dinero: ");
    scanf("%f",&cantdinero);
    while(mes<=12)
    {
        intereses = cantdinero * 0.02;
        cantdinero = cantdinero + intereses;
        printf("En el mes %d tiene en el banco %.2f\n",mes,cantdinero);
        mes++;
    }
    printf("\n\nDESPUES DE %d MESES-->TIENE EN EL BANCO:%f\n",mes-1,cantdinero);
}
```

Salida:



```
c:\ Console program output
Introduzca la cantidad de dinero: 5000
En el mes 1 tiene en el banco 5100.00
En el mes 2 tiene en el banco 5202.00
En el mes 3 tiene en el banco 5306.04
En el mes 4 tiene en el banco 5412.16
En el mes 5 tiene en el banco 5520.40
En el mes 6 tiene en el banco 5630.81
En el mes 7 tiene en el banco 5743.43
En el mes 8 tiene en el banco 5858.30
En el mes 9 tiene en el banco 5975.46
En el mes 10 tiene en el banco 6094.97
En el mes 11 tiene en el banco 6216.87
En el mes 12 tiene en el banco 6341.21

DESPUES DE 12 MESES-->TIENE EN EL BANCO:6341.208984
*** Process returned 54 ***
```

4.12 PROBLEMAS PROPUESTOS DE LA ESTRUCTURA DE CONTROL: while

1. Un Zoólogo pretende determinar el porcentaje de animales que hay en las siguientes tres categorías de edades: de 0 a 1 año, de mas de 1 año y menos de 3 y de 3 o mas años. El zoológico todavía no esta seguro del animal que va a estudiar. Si se decide por elefantes solo tomara una muestra de 20 de ellos; si se decide por las jirafas, tomara 15 muestras, y si son chimpancés tomara 40.
2. Encontrar el menor valor de un conjunto de **N** números dados.
3. Realice un programa en C que permita registrar las edades de los empleados de una fábrica (no se sabe cuántos) y verificar cuántos son mayores de edad (18 años o más) y cuántos tienen una edad menor a 18 años.
4. Realice un programa en C que calcule la suma de los números pares desde 1 hasta un número dado N.
5. Una persona debe realizar un muestreo con **20** personas para determinar el promedio de peso de los niños, jóvenes, adultos y viejos que existen en su zona habitacional. Se determinan las categorías con base en la sig, tabla:

<u>CATEGORIA</u>	<u>EDAD</u>
Niños	0 - 12
Jóvenes	13 - 29
Adultos	30 - 59
Viejos	60 en adelante

6. El Depto. de Seguridad Publica y Transito desea saber, de los **N** autos que entran a la ciudad de Nicaragua, cuantos entran con calcomanía de cada color. Conociendo el último dígito de la placa de cada automóvil se puede determinar el color de la calcomanía utilizando la sig. relación:

<u>DÍGITO</u>	<u>COLOR</u>
1 o 2	amarilla
3 o 4	rosa
5 o 6	roja
7 o 8	verde
9 o 0	azul