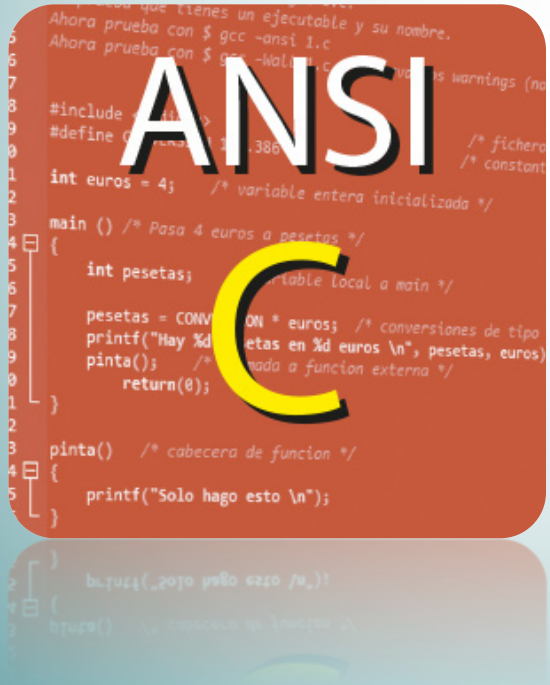
Programación Estructurada en ANSI C

Sesión 2B



Rafael Menéndez de Llano Rozas

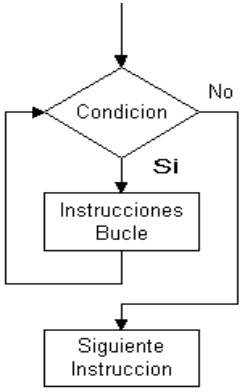
DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA

Índice

2. Iteración: Bucles

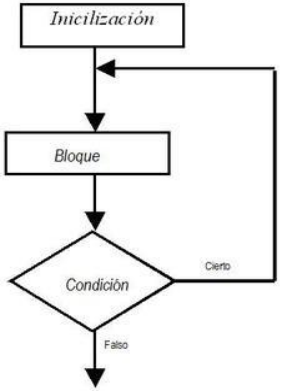
- Una sentencia de iteración o bucle se basa en repetir otra sentencia de acuerdo a una determinada condición.
- En general existen tres tipos de bucles:
  - Bucle **while**: No se sabe el número de iteraciones y la expresión se evalúa antes.

```
while ( condición )
      sentencia(s);
```



  - Bucle **do-while** (repeat - until): No se sabe el número de iteraciones y la expresión se evalúa después (se ejecuta al menos una vez).

```
do                              repeat
      sentencia(s);                   sentencia(s);
while ( condición );            until (condición);
```



  - Bucle **for**: se sabe el número de iteraciones y se evalúa antes.

```
for (inic; cond ; next)     for i=ini to fin (step)
    sentencia(s);                   sentencia(s);
```

2. Comparación de iteración

Java	C

`while` (exp_booleana) {
  instrucciones;
}
	`while` (exp_entera) {
	  instrucciones;
	}

`while` (true) {
  instrucciones;
}
	`while` (1) {
	  instrucciones;
	}

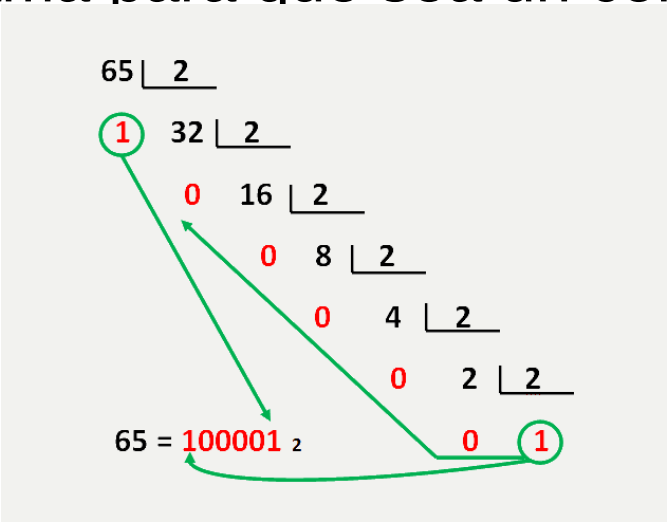
`do` {
  instrucciones;
} `while` (exp_booleana);
	`do` {
	  instrucciones;
	} `while` (exp_entera);

Java	C

`for` (`int` i=v_inic; i<=v_fin; i++)
{
  instrucciones;
}
	`for` (i=v_inic; i<=v_fin; i++)
	{
	  instrucciones;
	}

Ejercicio 10: Sentencias de iteración

1. Bájate el programa 11.c y guárdalo en tu directorio.
2. Ábrelo con un editor y examínalo y comprende las operaciones que hay.
3. Introduce varios pares de números para ver que se dan diversas vueltas.
4. Cambia el programa para que sea un conversor de base.



2. Iteración: while

```
#include <stdio.h>
int main()
{
    int dividendo, divisor, resto;
    int contador = 0;
    printf ("Introduce dividendo y divisor : \n");
    scanf("%d %d", &dividendo, &divisor);
    while (dividendo >= divisor)   /* solo sigo si es mayor */
    {
          resto = dividendo % divisor;     /* hago la division */
          dividendo = dividendo / divisor;   // /=
          printf("%d %5d | ", dividendo, resto);       /* a la derecha */
          contador++;      /* aumento contador de divisiones */
    }
    printf ("\Lo que queda es %d  ", dividendo);
    printf("El numero de divisiones : %d \n", contador);
    return 0;
}
```

Ejercicio 11: Sentencias de iteración

1. Bájate los programas 12.c y 13.c
2. Guárdalos en tu directorio.
3. Se trata de sumar los números introducidos hasta que se escriba uno negativo.
4. Date cuenta que si empiezas por uno negativo el do–while no vale.

2. Iteración: while / do-while

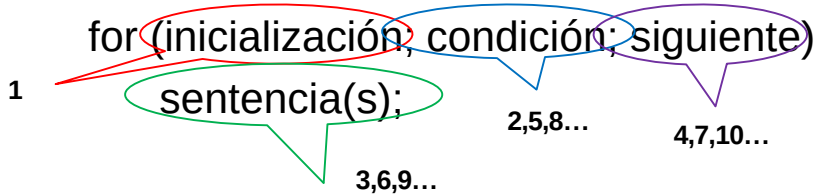
```
#include <stdio.h>
int main()
{
    int numero, suma = 0;
    scanf("%d", &numero);
    while (numero >= 0)
    {
          suma += numero;
          scanf("%d", &numero);
    }
    printf("La suma es : %d\n", suma);

    return 0;
}
```

```
/* cambialo para que funcione */
#include <stdio.h>
int main()
{
    int numero, suma = 0;
    do {
          scanf("%d", &numero);
          suma += numero;
    }
    while (numero >= 0);
    printf ("La suma es : %d\n",
    suma);
    return 0;
}
```

2. Iteración: for

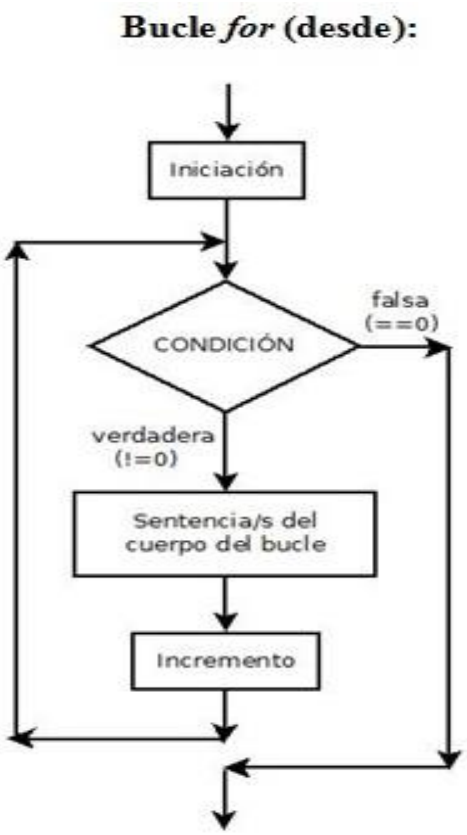
- Bucle **for**: "se sabe" el número de iteraciones y se evalúa antes:



- El bucle **for** es también como un while encubierto:

```
cont = 1;
while (cont <= numero)
{
   printf("hola\n");
   cont++;
}
```

```
for (cont =1; cont <= numero; cont++)
  printf("hola\n");
```



Ejercicio 12: Sentencias de iteración

- Debido a su gran flexibilidad, existen muchas maneras de utilizarlo:

1. Bájate el programa 14.c y guárdalo en tu directorio.
2. Ábrelo con un editor, examínalo y comprende las operaciones que hay dentro de los bucles for.
3. Cambia algunos de los ejemplos.

2. Sentencias de control

- **continue**: para salir de una iteración (bucles).

```
while ((ch = getchar()) != EOF)
{
        if (ch == ‘a')
                continue;
        putchar(ch);
}
```

- **break**: en bucles se sale del todo.

```
while ((ch = getchar()) != EOF)
{
        if (ch == ‘a')
                break;
        putchar(ch);
}
```

Ejercicio 13: Sentencias de control

- Bájate los programas 15.c y 16.c guárdalos en tu directorio.

1. Ábrelos con un editor, examínalo y comprende las operaciones que hay dentro del bucle `while`.
2. Indica cuando terminará el programa 15.
3. Indica que aparecerá y que no, cuando ejecutes el programa 16.
4. ¿Es el 16 una algoritmo propio?

Observación: Estas dos sentencias son equivalentes al goto (de forma encubierta), en la programación estructurada está denostado el uso de esta sentencia, ya que puede ser sustituida por otras que harán los programas más claros. En mi opinión sólo son justificables en casos de detección de error y acompañadas de una condición.