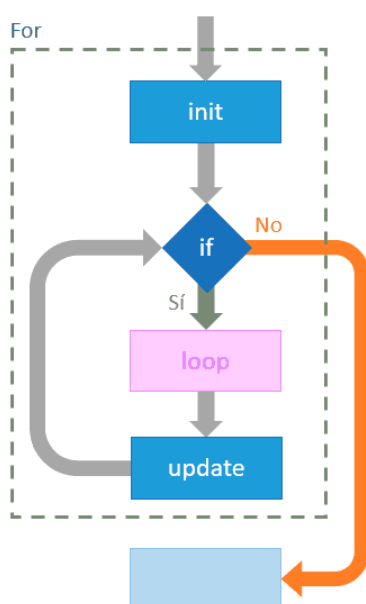


### Explicació

El bucle FOR és una estructura de control que permet repetir un bloc de codi un nombre específic de vegades. És una de les estructures de control més utilitzades i freqüents.

A mesura que s'utilitzaven els bucles, es donava molt sovint el cas que es necessitava fer alguna cosa N vegades (100, 1.000, 1.000.000), sense dependre d'una condició.

Tot el que podem fer amb un bucle for ho podem fer amb un bucle while, però ho podem veure d'una forma un poc més clara.



La sintaxi d'un bucle for en C# es compon de tres parts principals: **inicialització, condició i actualització**.

Aquestes parts es detallen dins dels parèntesis del bucle for i estan separades per punt i coma.

L'estructura bàsica d'un bucle for en C# és la següent:

```
1 for(initialization; condition; update)
2 {
3     // Instructions
4 }
```

En aquest bucle FOR, tal com està definit en molts llenguatges, consta de tres parts.

- **initialization:** una sentència que s'executa abans d'entrar en el bucle.
- **update:** sentència que s'executa a cada iteració del bucle.
- **condition:** sentència que s'avalua per continuar el bucle.

Equival amb un WHILE a:

```
1 initialization;
2 while(condition) {
3     // Instructions
4     update;
5 }
```

**RECORREGUTS SIMPLES AMB FOR**

1. Escriure els 20 primers números en ordre ascendent.

1, 2, 3,..... 20

2. Escriure els 20 primers números parells.

2, 4, 6,..... 40

3. Fes un programa que demani dos números i mostri per pantalla tots els valors compresos entre aquests dos números.

ENTRA n1: 5

ENTRA n2: 900

5, 6, ....., 900

4. Modifica l'anterior programa fent que no es mostrin els números que siguin múltiples de 7.

5. Fes un programa que demani 10 números per teclat i que els sumi. Al final del recorregut ha de mostrar la suma de tots els números.

6. Fes un programa que demani 10 números per teclat. En acabar el programa ha d'informar de quants números son positius, negatius i zeros

7. Fes un programa que demani per teclat un número i calcule la suma del número 1 al número entrat per l'usuari.

ENTRA UN NÚMERO: 4

10 //1+2+3+4

8. Fes un programa que demani per teclat un número i calcule el producte del número 1 al numero entrat per l'usuari.

ENTRA UN NÚMERO: 4

24 //1\*2\*3\*4

NOTA: El producte d'un número ... es coneix amb el nom del factorial.

9. Escriu un programa que demani un número per teclat. El programa ha de mostrar la taula de multiplicar de l'1 al 10 del número entrat per teclat.

ENTRA UN NÚMERO: 2

$2 \times 0 = 0$

$2 \times 1 = 2$

...

$2 \times 10 = 20$

10. Fes un programa que mostri tots els números múltiples de 4 entre el 0 i el 300.

- Després de mostrar 20 números, el programa ha de pausar-se fins que l'usuari premi una tecla.
- Utilitza la instrucció **Console.ReadKey()** per esperar que l'usuari entri una tecla.

**RECORREGUTS DE FITXERS AMB FOR**

11. Considerem el fitxer **BONUS3.TXT** per al càlcul del BONUS. La primera línia del fitxer conté el nombre total de bitllets. Cada línia només informa de si el bitllet té un "BONUS" o no. Si la línia té un "BONUS", la següent línia contindrà el valor "BONUS", i en cas contrari, "NO BONUS". Fes un programa que processi un fitxer de dades que contingui línies de "BONUS/NO BONUS" en el format indicat i informi de quants bonus s'han distribuït, quants bitllets hi apareixen en el fitxer i quin percentatge de bitllets guanyadors hi ha respecte al total. Ha d'indicar també quin és el guany obtingut amb els bonus trobats al fitxer.

12. Considerem el fitxer **coordenades2.txt** on en la primera línia trobarem la quantitat de coordenades que té el fitxer i la resta contenen les coordenades de diferents punts amb el següent format: cada coordenada x a cada línia senar i cada coordenada y a cada línia parell. S'introdueix per teclat el valor del radi r d'una circumferència.

Un exemple de fitxer amb el format especificat anteriorment:

-2,6 → coordenada x

3,8 → coordenada y

...

El programa ha d'informar per a cada punt si està dins, fora o sobre el perímetre de la circumferència de radi r i centre (0.0).

Crea i utilitza la funció *Distancia* que retorna la distància d'un punt a l'eix de coordenades.

**public static double Distancia(double x, double y)**

13. Considerem el fitxer **Girona lliga23\_24\_v2.txt** que conté resultats del Girona de la temporada 23/24 en un fitxer CSV. Realitza el mateix exercici que férem als recorreguts sabent que la primera entrada ens dirà el total de jornades de la lliga.

14. Donat el fitxer **tiquetsuper.txt**, calcular la el preu total de la compra del tiquet. La primera línia del fitxer indica el nombre de productes. Després cada producte està format per tres línies (Nom del producte, Quantitat comprada de producte, Preu unitari del producte). El programa haurà d'indicar el preu final del tiquet.

**tiquetsuper.txt**

4

NATURAL YOGURT → nom del producte

8 → quantitat de producte

0,5 → preu del producte

...

**BUCLES ANIUATS**

15. Fes un programa que demani a l'usuari un número i que mostri un quadrat.

ENTRA UN NÚMERO: 4

```
****  
****  
****  
****
```

**Opcional**

Utilitzar la funció **GenerarLinia**, que genera una cadena de text d'una longitud específica amb un caràcter donat.

**public static string GenerarLinia(char car, int longitud)**

16. Fes un programa que demani a l'usuari un número i que mostri una piràmide.

ENTRA UN NÚMERO: 4

```
*  
**  
***  
****
```

**Opcional**

Utilitzar la funció **GenerarLinia**, que genera una cadena de text d'una longitud específica amb un caràcter donat.

**public static string GenerarLinia(char car, int longitud)**

17. Fes un programa que mostri per pantalla totes les taules de multiplicar del 0 al 10.

**Opcional**

Utilitzar el procediment **MostrarTaula**, que mostri per pantalla la taula de multiplicar d'un número passat per paràmetre.

**public static void MostrarTaula(int numTaula)**

18. Fes un programa que mostre una piràmide de números com la següent

```
  1
 1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

19. Fes un programa que demani un número enter positiu  $n$  i mostri el següent ( exemple per  $n=4$ )

$1=1$

$1+2=3$

$1+2+3=6$

$1+2+3+4=10$