



Composició iterativa

Objectius:

- Practicar la composició iterativa (`while` i `for`).
- Construir una iteració.
- Entendre el concepte d'*overflow*.
- Executar un programa amb l'entrada en un fitxer.
- Calcular la mitjana i el màxim d'una seqüència.
- Executar un programa i guardar la sortida a un fitxer.
- Saber fer un programa amb un doble bucle.

(Objectius: 1.1.8, 1.5.3, 1.5.4, 1.5.6, 1.5.7, 1.5.8)

(Documents relacionats: [Introduccio_redireccio.pdf](#))

1. Bart Simson

Com és sabut, Bart Simpson, de la família Simpson, és porta molt malament. És típic que li facin escriure en un paper 100 vegades: "No em tornaré a portar malament", o "No tornaré a parlar a classe", etc. Imagina, però, que Bart Simpson té un ordinador i pot fer un programa com el següent:

```
#include <iostream>
using namespace std;

int main()
{
    int i = 1;
    while ( i < 100 ) {
        cout << i << ". No tornaré a parlar a classe" << endl;
    }
}
```

Un cop entrat el programa, Bart Simpson l'executa, però s'adona que alguna cosa falla. Quantes vegades es mostrarà la frase en aquest programa?

Com pots arreglar-lo perquè la frase surti 100 vegades, exactament?

2. Nombres

Fes un programa que visualitzi tots els múltiples de 3 entre 1 i 1000 i dona també la suma de tots ells.

3. Lletres

Fes un programa que demani 2 lletres de l'abecedari (minúscules) i mostri per pantalla totes les lletres intermitges. Per exemple:

Entra la primera lletra: **a**
Entra la última lletra: **m**
abcdefghijklm

La part en taronja és el que se suposa que l'usuari del programa escriurà.

4. Parells-senars del rang

Implementeu un programa que visualitzi tots els parells entre M i N (límits inclosos) i la mitjana dels senars. Els valors M i N els introduirà l'usuari per teclat. Els valors M i N poden estar en ordre creixent, decreixent o iguals.

5. Taula ASCII

Visualitza la taula [ASCII](http://es.wikipedia.org/wiki/Código_ASCII) (http://es.wikipedia.org/wiki/Código_ASCII) des del valor 32 al 126. Només cal visualitzar el valor numèric i el caràcter corresponent. Exemple: 65 A.

6. Potències

Examina el programa següent i intenta veure què fa:

```
#include <iostream>
using namespace std;

int main()
{
    int i = 1, n, pot = 1;
    cin >> n;
    while ( i < n ) {
        pot = pot * 10;
        i++;
    }
    cout << pot << endl;
}
```

Si poses com a entrada 2, què sortirà per pantalla? I si poses 7?

7. Doblegar un paper

Es sabut que no es pot doblegar un full de paper comú més de 9 vegades sobre si mateix. Fes un programa que demani a l'usuari el número de vegades que es vol doblegar el paper i es farà el càlcul per un full té un gruix de 0,05 mm. Mira què passaria si es pogués doblegar 100 vegades.

(Us dono unes dades que us permetrà reflexionar sobre el resultat:

1 Any Llum = 9.461.000.000.000 km. Distància a la Terra de la Constel·lació de Orió: 1.500 Anys Llum)

8. Factorial

El factorial d'un nombre **n** (que s'escriu **n!**) és la multiplicació de tots els nombres des de 1 fins a **n**. O sigui:

$$6! = 1 * 2 * 3 * 4 * 5 * 6 = 720$$

Fes un programa que llegeixi un número a l'entrada i després calculi el factorial d'aquest número. A partir de quin número el programa calcula el factorial malament? A què penses que es deu?

9. Número de xifres

Donat un nombre, fixa't que si el divideixes per 10, de fet li "treus una xifra". Fes un programa que demana un número a l'usuari i després li diu quantes xifres té.

10. Una suma

Imagina't que algú et dona un fitxer de text (amb extensió **txt**) i el fitxer conté una llista de números (amb un -1.0 al final). Podem fer un programa que calculi la suma d'aquests números? La resposta és que sí, però requereix que executem el programa des de una finestra de MS-DOS. Primer entra el següent programa (**suma.cpp**):

```
#include <iostream>
using namespace std;

int main()
{
    float x, suma = 0.0;
    cin >> x;
    while ( x != -1.0 ) {
        suma = suma + x;
        cin >> x;
    }
    cout << "Resultat: " << suma << endl;
}
```

Ara compila el programa. Ves a Atenea i descarrega el fitxer **dades1.txt** que veuràs a la carpeta jocs de prova. El guardes a la carpeta a on està el programa. Obre un terminal (una finestra MS-DOS) i mou-te cap la carpeta (o directori) a on hi ha el programa. Ara executa el programa així:

```
H:\FI\sessio5> suma < dades1.txt
```

El que has d'escriure tu és la part taronja. Aquesta comanda executa el programa **suma** (recorda la sessió 1!), però el que passa és que quan el programa fa un "**cin >> ...**", en comptes de treure les dades del teclat (i esperar que pitgis Return), les va a buscar directament al fitxer, o sigui que el fitxer és converteix en l'entrada del programa. Així, el programa no mostra res per pantalla excepte el resultat:

```
Resultat: 17.643
```

Executar un programa d'aquesta manera té un avantatge gran, i és que per dades d'entrada llargues, no cal que les escriguis una per una. Per altra banda, pots provar el funcionament d'un programa de forma

ràpida, també.

Amb el mateix programa, calcula ara quin és el resultat pels fitxers de dades `dades2.txt` i `dades3.txt` que també trobaràs a Atenea. Tots dos fitxers contenen una llista de números amb un `-1.0` al final.

11. La mitjana

Modifica el programa anterior per tal que calculi la mitjana de les dades i calcula la mitjana per als tres fitxers `dades1.txt`, `dades2.txt` i `dades3.txt`. Compara el resultat que et dona amb el dels teus companys.

12. El màxim

Crea un programa per calcular el màxim valor de una llista de números acabada en un `-1.0` (tal com fins ara). Aplica'l a calcular el màxim dels tres fitxers `dades1.txt`, `dades2.txt` i `dades3.txt`.

13. Les taules de multiplicar

Fes un programa que demani un número i mostri la taula de multiplicar d'aquest número. Per exemple:

```
Entra un número: 7
1 * 7 = 7
2 * 7 = 14
3 * 7 = 21
4 * 7 = 28
5 * 7 = 35
6 * 7 = 42
7 * 7 = 49
8 * 7 = 56
9 * 7 = 63
10 * 7 = 70
```

Com sempre, la part taronja és el que ha escrit l'usuari.
En aquest programa, en comptes d'utilitzar el `while`, utilitza el `for`.

14. Doble bucle

Modifica el programa anterior, per tal que es mostrin per pantalla totes les taules de multiplicar (de l'1 al 10). Aquest programa es dirà `taules_mult.cpp`. Posa una línia buida entre cada taula i la següent. Un cop tinguis el programa compilat i vegis que funciona, obre una finestra de MS-DOS i executa el següent:

```
H:\FI\sessio4> taules_mult > les_taulas.txt
```

Això invoca el teu programa, però ara (fixa't que el signe és el major, no com abans) la sortida del programa anirà a parar al fitxer `les_taulas.txt`. Teòricament, quan executis el programa així, per pantalla no sortirà res, anirà tot al fitxer directament. Obre el fitxer amb el "Notepad" o amb Dev-C++ mateix per veure'n el contingut i comprova com les taules de multiplicar estan allà.

Altres exercicis:

1. Dibuixa quadrat

Fes un programa que demani un valor enter i visualitzi un quadrat amb aquest valor de costat. El dibuix es farà amb el símbol *. Exemple de costat 3:

* * *

* * *

* * *

2. Nombres perfectes

Fes un programa que determini si un nombre enter introduït per teclat és perfecte. Busca que s'entén per nombre perfecte.

3. Dilema del presoner

Demaneu les indicacions al vostre professor per fer aquest exercici.