



Composició alternativa

Objectius:

- Practicar la composició alternativa (`if`).
- Saber què és la indentació i com indentar un programa.
- Entendre l'ús de les claus ('{' i '}').
- Detectar repetició en les alternatives d'un `if`.
- Construir una expressió per parts.
- Saber fer un programa amb més d'una alternativa.

(Objectius: 1.5.4, 1.5.5)

[\(Documents relacionats: Indentació.pdf\)](#)

Exercicis participatius: 2,3,4

1. El màxim i el mínim

Compila i executa el següent programa:

```
#include <iostream>

using namespace std;

int main()
{
    float a, b;

    cin >> a >> b;

    if ( a > b ) {

        cout << "El màxim és: " << a << endl;

    }
```

```

else {

    cout << "El màxim és: " << b << endl;

}

}

```

Compara'l ara, amb aquest altre:

```

#include <iostream>

using namespace std;

int main()

{

    float a, b, max;

    cin >> a >> b;

    if ( a > b ) max = a;

    else max = b;

    cout << "El màxim és: " << max;

}

```

Comprova que fan el mateix i que entens quines diferències tenen.

Modifica aquest últim per tal que calculi el mínim.

2. Parell-Senar

Fes un programa que demani un número sencer i visualitzi si el nombre és parell o senar.

3. Àrea Figura

Fes un programa que permeti calcular l'àrea d'un triangle o d'un rectangle segons decideixi l'usuari. Per fer això l'usuari introduirà la lletra *R* si vol l'àrea del rectangle o la *T* si vol la del triangle. A continuació es

demanarà la base i alçada de la figura i s'haurà de visualitzar l'àrea de la figura demanada.

4. Indentació

Fixa't en els programes que hem fet fins ara: tots estan *ben indentats*. La indentació és el número d'espais que es posen al principi d'una línia. Sabent això, corregeix la indentació del següent programa:

```
#include <iostream>

using namespace

std

; int

main( ){

int a; int b; cin >> a;

cin >> b; if ( a < b ) { b = b -10; a =1;

} else b=a

;cout

<< "Els valors són:" <<

a << b; cout << endl; }
```

Una bona indentació és important a l'hora de fer programes ja que altera moltíssim la facilitat de comprensió. El compilador no té problemes per entendre el codi mentre es respecti la sintaxi al peu de la lletra, però per llegir el programa és important que estigui ben indentat, i al final resulta que els programes els llegim més les persones que no pas els ordinadors.

5. Blocs

Considera el següent programa:

```
#include <iostream>

using namespace std;
```

```

int main()

{

    int a, b, tmp;
    cin >> a >> b;
    if ( a > b ) {
        cout << "Major!" << endl;
        tmp = a;
        a = b;
        b = tmp;
    }
    cout << a << " < " << b;
}

```

Primer pensa què fa el programa. Ara esborra les claus, marcades amb taronja i torna a executar-lo.

Què observes? Quina conclusió es pot treure del resultat?

6. Dia de la setmana

Fes un programa que demani un número (el dia de la setmana, començant per 1) i escrigui per pantalla el nom del dia ("dilluns", "dimarts", etc.). Fes, a més a més, que el dia 0 sigui també diumenge (és a dir, que surti "diumenge" tant si poses 0 com 7).

7. Triangles

Implementa un programa que demani 3 longituds (els costats d'un triangle) i després digui si el triangle és:

- **Equilàter:** tots els costats són iguals.
- **Isòsceles:** 2 dels costats són iguals.
- **Escalè:** tots els costats són diferents.

El programa simplement ha de mostrar per pantalla quin d'aquests 3 tipus és.

8. Repetició

Si en programa hipotètic tenim un `if` com el següent:

```

if (a != b) {

    cout << "Diferents" << endl;
    a = a + b;

    c = b;

}

```

```

else {

    cout << "Iguals" << endl;

    a = a + b;

    c = b;

}

```

fixa't que les dues últimes instruccions (en color verd) de cada branca de l'`if` són les mateixes. Per aquest motiu, no cal que estiguin repetides a cada branca, les podem posar després:

```

if ( a != b )

    cout << "Diferents" << endl;

else

    cout << "Iguals" << endl;

a = a + b;

c = b;

```

Fixa't que, a més, com que només hi ha una instrucció, un cop arreglada la repetició, no calen les claus, ja que només queda una instrucció a cada branca.

Observa ara aquest altre `if`:

```

if ( a == b ) {

    c++;

}

if ( a == b && c == 10) {

    c = 0;

}

```

La condició del segon `if` és una "ampliació" de la del primer (és més restrictiva, però comença igual). Això es pot arreglar així:

```
if ( a == b ) {  
    c++;  
    if ( c == 10 ) c = 0;  
}
```

Això és perquè si posem el segon `if` a dintre del primer, es complirà que `a == b`, per tant el segon compleix les dues condicions (o sigui, afegeix `c == 10`).

Tenint en compte tot això, doncs, corregeix el següent programa:

```
#include <iostream>  
  
using namespace std;  
  
int main()  
{  
    int hora, minut;  
    cin >> hora >> minut;  
    minut = minut + 1;  
  
    if ( hora = 23 && minut = 59 ) {  
        minut = 0;  
        hora = 0;  
    }  
  
    if ( minut = 59 ) {  
        minut=0;  
        hora = hora + 1;  
    }  
}
```

```

    }

    cout << "Són les: " << hora << ":" << minut << endl;

}

```

Aquest programa també té un **error** que passa una mica desapercebut, el veus?

Revisa els exercicis 4 i 5 comprova si has repetit algun bloc.

9. Relotges

Implementeu un programa que visualitzi l'hora per diferents ciutats. L'usuari introduirà hora i minuts i es visualitzarà la [correspondència horària](http://es.wikipedia.org/wiki/Huso_horario#UTC.2B5) (http://es.wikipedia.org/wiki/Huso_horario#UTC.2B5) per Hong-Kong (+7), New York (-6) i San Francisco (-9).

10. Dies del mes, versió limitada

Fes un programa que demani un número que representa un mes (de l'1 al 12), i llavors digui quants dies té aquest mes (mostri per pantalla el número de dies). En aquesta versió, els mesos poden tenir 30, 31 o 28 dies (no tinguis en compte de moment els anys de traspàs).

11. Major de 3

Fes un programa que demani tres números i visualitzi el més gran dels 3.

12. Anys de traspàs

En aquest exercici construirem una expressió Booleana que s'avalui a **true** si un any **a** és de traspàs (quan Febrer té 29 dies) i **false** si no ho és. Això ho farem per parts. Primer entra el següent programa:

```

#include <iostream>

using namespace std;

int main()

{

    int a;

```

```

cout << "Entra l'any: ";

cin >> a;

if ( expressió ) {

    cout << "De traspàs!" << endl;

}

else {

    cout << "Normal" << endl;

}

}

```

La part en taronja indica a on anirà l'expressió per a l'any de traspàs. Els passos a seguir són, doncs:

1) Els anys de traspàs són, en general, els múltiples de 4. Pensa una expressió primer que indiqui si un número és múltiple de 4 i la poses en el programa, tot comprovant que funcioni amb diversos números. Comprova també com en l'any 100, es mostra per pantalla "De traspàs!" (ja que 100 és múltiple de 4).

2) Ara bé: els múltiples de 100 de fet **no** són de traspàs. Modifica ara l'expressió de tal manera que just en els múltiples de 100, el programa digui "Normal", no com abans. Has d'ajuntar una expressió a l'anterior per tal d'aconseguir això (pensa primer una expressió que digui que un número *no* és múltiple de 100).

3) Finalment, resulta que els anys que just són múltiples de 400 **sí** que són de traspàs. Llavors, 1900 ha de donar "Normal", 1904 "De traspàs!" i 1600 "De traspàs!". Pensa com "afegir" els anys múltiples de 400 a l'expressió que tens feta fins ara. Comprova que tot funcioni.

13. Dies del mes per a cert any

Modifica el programa de l'exercici 10 per tal que es doni el número de dies de cada més, però tenint en compte els anys de traspàs (o sigui, l'any 2000, Febrer tenia 29 dies). Ara el programa demanarà 2 valors: l'any (per saber si Febrer té 29 dies), i el mes. Per fer-ho combina el codi dels dos programes anteriors (ex. 10 i ex. 12). Intenta primer fer que el programa funcioni, i seguidament eliminar la repetició (tal com has vist en l'exercici 4).

Altres exercicis:

1. Escacs

Fes un programa que determini si una casella del taulell d'escacs és blanca o negra. L'usuari haurà d'introduir la fila i la columna i es visualitzarà un missatge indicant el color.

2. Bitllets de banc

Fes un programa que trobi el menor nombre de bitllets de banc de 100, 50, 20, 10 i 5€ necessaris per pagar una quantitat introduïda per l'usuari.

3. Suma de valors

Fes un programa que demani 3 valors numèrics i determini si un d'ells és la suma dels altres 2.

4. Trucada telefònica

Fes un programa per calcular la quantitat a pagar per una trucada telefònica d'acord amb les següents premisses:

- Una trucada amb duració inferior o igual a 3 minuts té un cost de 20 cèntims
- Cada minut addicional té un cost de 5 cèntims

5. Notes

Fes un programa per fer la conversió d'una nota numèrica (amb decimals), introduïda per teclat, a la seva corresponent nota alfanumèrica segons els valors habituals (I: insuficient, S: suficient, B: bé, N: notable, E: Excel·lent).

6. Compara dates

Fes un programa que llegeixi dues dates (dia, mes i any de cadascuna) i digui si la primera és anterior en el temps a la segona.

7. Formen triangle

Fes un programa que donades les longituds dels costats d'un triangle determini si amb aquests valors es pot construir un triangle. *Nota: Es pot construir un triangle si la suma de qualsevol parell de costats és superior al tercer (i això és així per tots els costats).*