

**Objectius de la pràctica:**

- Conèixer els operadors relacionals i lògics.
- Utilització de les estructures de control selectives en la realització de programes.

**Operadors relacionals**

Els operadors relacionals o de comparació s'empren per a comparar dos quantitats i tornen un valor "VERTADER" si la relació és certa o un valor "FALS" si no ho és. Ni hi ha cap altra possibilitat.

Les quantitats que es comparen poden ser: constants, variables, valors retornats per funcions o expressions en què es combinen totes tres. El conjunt de les quantitats combinades amb els operadors relacionals formen una expressió relacional.

Els operadors relacionals definits en el llenguatge C són:

> Major que	!= Distint de	>= Major o igual que	<= Menor o igual que
< Menor que	== Igual que (no confondre amb l'operador d'assignació =)		

**Sobre els valors VERTADER i FALS en C/C++**

En el llenguatge C, qualsevol valor distint de zero es considera VERTADER mentre que sols serà FALS quan siga igual a zero. Així, doncs, l'expressió condicional `if (x)` té sentit en C, ja que serà certa sempre que `x` siga diferent de zero, siga quin siga el tipus de `x`.

D'una altra banda, havíem dit adés que tota expressió relacional tornava un valor de VERTADER o FALS. En realitat, s'avalua com a 1.0 quan l'expressió és certa o 0.0 quan és falsa.

**Operadors lògics**

&& AND lògic	OR lògic	! NOT lògic
--------------	----------	-------------

**Prioritat i ordre d'avaluació dels operadors aritmètics, lògics i relacionals**

1) ( )	2) - ! ++ --	3) * / %	4) + -	5) < <= > >=
6) == !=	7) &&	8)	9) = *= /= %= += -=	

**Sentències de Control Selectives**

- SENTÈNCIA `if` (Alternativa senzilla)

Pren una decisió sobre una sentència (o grup de sentències) que s'ha d'executar a continuació, en funció del resultat (vertader o fals) d'una expressió.

<b>Sintaxi:</b>	
<code>if (Expressió Condicional)   Sentència_1;</code>	<code>if (Expressió Condicional) {   Sentència_1;   Sentència_2;    Sentència_n; }</code>



L'expressió condicional està continguda entre parèntesis i el seu resultat decideix si la Sentència\_1 (o el bloc de sentències {Sentència\_1; Sentència\_2; ...; Sentència\_n; }), que ve a continuació, s'ha d'executar o no (només s'executarà quan siga VERTADERA).

El flux del programa continuarà sempre cap a endavant, tant si s'ha executat la Sentència\_1 (o el bloc de sentències) com si no, amb la resta del programa.

- SENTÈNCIA `if ... else` (Alternativa doble)

Ofereix la possibilitat d'executar una o més sentències alternatives en el cas que l'expressió condicional s'avalue com a FALSA.

Sintaxi:	
<pre>if (Expressió Condicional)     Sentències_1; else     Sentències_2;</pre>	<pre>if (Expressió Condicional) {     Bloc de sentències 1; } else {     Bloc de sentències 2; }</pre>

Si el resultat de l'expressió és vertader s'executarà el bloc de sentències 1 i, si és fals, el bloc de sentències 2.

El flux del programa transcorre sempre cap a endavant. Açò és, el programa executarà només un dels dos blocs de sentències i, a continuació, executarà el que resta de programa.

### Imbricació de sentències `if ... else`

Com a part del bloc de sentències 1 o 2 poden aparèixer altres sentències `if ... else`. El bloc de sentències `if ... else` és un cas particular d'imbricació de sentències `if ... else` que convé conèixer, ja que és una configuració que es troba molt a sovint dins dels programes.

Sintaxi:
<pre>if (Expressió Condicional 1) {     Bloc de sentències 1; } else if (Expressió Condicional 2) {     Bloc de sentències 2; } ... else if (Expressió Condicional n-1) {     Bloc de sentències n-1; } else {     Bloc de sentències n; }</pre>



Les expressions condicionals s'avaluaran de dalt a baix i, quan es trobe la primera d'aquelles el valor de la qual siga VERTADER, el programa executarà el seu corresponent bloc d'instruccions. Si no es compleix cap de les condicions s'executarà el bloc d'instruccions associat al darrer else (en el cas que existisca).

- SENTÈNCIA switch (Alternativa múltiple)

Permet executar una entre un conjunt d'accions en funció del valor d'una expressió entera (short, int o long) o de tipus caràcter (char).

**Sintaxi:**

```
switch (expressió)
{
    case Constant_1:
        Bloc de sentències 1;
        break;
    case Constant_2:
        Bloc de sentències 2;
        break;
    case ...
        ...
    default:
        Bloc de sentències per defecte;
}
```

La sentència switch compara successivament el valor de l'expressió continguda entre parèntesis amb cadascuna de les constants situades darrere de la paraula case. Quan troba una correspondència, executa el conjunt de sentències que ve a continuació fins a trobar una sentència break o fins que acaba la sentència switch. La instrucció break és una ordre imperativa d'eixida del bloc de la sentència switch.

Com en els casos anteriors, el flux del programa transcorre cap a endavant i, en acabant d'executar el conjunt de sentències associat a la constant que coincideix amb l'expressió, l'execució del programa continuarà amb la resta d'instruccions que vinguen a continuació.

**Errors comuns de programació**

1. Un dels errors més freqüents en una sentència if és l'ús de l'operador d'assignació '=' en comptes de l'operador de comparació '=='.
2. En una sentència if imbricada, cada clàusula else es correspon amb la instrucció if precedent més propera. Per exemple, en el codi següent:

```
if (a > 0)
if (b > 0)
c = a + b;
else
c = a * b * c;
d = a * b;
```

Quina és la sentència if associada a l'else?



La manera més senzilla d'evitar errors és el sagnat, amb el qual és fàcil veure com la clàusula `else` es correspon amb la sentència que conté la condició: `b > 0`

```
if (a > 0)
    if (b > 0)
        c = a + b;
    else
        c = a * b * c;
d = a * b;
```

Una de les raons per emprar la Guia d'Estil és la claredat dels programes, fet que permetrà evitar errors innecessaris.

3. El selector d'una sentència `switch` ha de ser de tipus enter o compatible (ordinal). Per tant, les constants reals (com ara 2.4, -4.5 o 3.1416) no poden ser emprades darrere d'un `case`.
4. Quan es fa servir una sentència `switch`, assegureu-vos que el selector de `switch` i les etiquetes `case` són del mateix tipus (`int`, `char`, però no `float`). Si el selector pren un valor que no estiga especificat en cap de les sentències `case`, la sentència `switch` no gestionarà cap acció. Per a solucionar aquest problema, convé afegir una etiqueta `default`.

### **Resum de CONCEPTES FONAMENTALS: Sentències IF i SWITCH**

Sentència if	Sentència switch
<pre>// Una alternativa if (a != 0)     resultat = a / b;  // Dues alternatives if (a &gt;= 0)     cout &lt;&lt; "positiu"; else     cout &lt;&lt; "negatiu";  // Alternatives múltiples if (x &lt; 0) {     cout &lt;&lt; "negativo";     abs_x = -x; } else {     if (x == 0)     {         cout &lt;&lt; "zero";         abs_x = 0;     }     else     {         cout &lt;&lt; "positiu";         abs_x = x;     } }</pre>	<pre>switch (nota) {     case 'a':         cout &lt;&lt; "Excel·lent";         break;      case 'b':         cout &lt;&lt; "Notable";         break;     case ...      case 'c':         cout &lt;&lt; "Aprovat";         break;     case ...      case 'd':         cout &lt;&lt; "Suspens";         break;      default: // altres casos         cout &lt;&lt; "nota no vàlida"; } // final del switch</pre>