

Algoritmi cu structură alternativă (recapitulare)

-**structura alternativă** permite programului ca să aleagă între două sau mai multe grupuri de instrucțiuni care să fie executate *în funcție de îndeplinirea unei condiții (expresie logică)*.

Dacă *condiție* atunci
 instrucțiuni1

altfel
 instrucțiuni2

Sfârșit dacă

Dacă *condiție* atunci
 instrucțiuni1

Sfârșit dacă

- o **propozitie logică** are doua valori : *adevarat(1), fals (0)*.
- propozițiile logice pot conține **operatori relaționali** : < , <= , > , >= , == egal , != (diferit)
- **expresiile logice** sunt formate din una sau mai multe propozitii combinate cu **operatorii logici** AND (și) , OR (sau), NOT (negația).

Structura alternativă în C++ (pag. 100-102)

- operatorii logici în C++ se notează : **&&** (și logic), **||** (sau logic), **!** (negația logică)

propozitia a	propozitia b	a && b
adevarat (1)	adevarat (1)	adevarat (1)
adevarat (1)	fals (0)	fals (0)
fals (0)	adevarat (1)	fals (0)
fals (0)	fals (0)	fals (0)

propozitia a	propozitia b	a b
adevarat (1)	adevarat (1)	adevarat (1)
adevarat (1)	fals (0)	adevarat (1)
fals (0)	adevarat (1)	adevarat (1)
fals(0)	fals(0)	fals(0)

propozitia a	!a
adevarat (1)	fals (0)
fals (0)	adevarat (1)

- exerciții : Care sunt rezultatele expresiilor logice următoare pentru a =5, b = 7 ?

(a>4) && (b<8)

(a>4) || (b>8)

- instrucțiunea if (dacă)

if (expresie logică)

```
    i1 ;  
[else  
    i2 ; ]
```

i1, i2 poate fi un grup de instrucțiuni scris între { }

- exemplu :

```
int n ;  
cin >> n ;  
if (n%2 == 0)  
    cout << "numar par" ;  
else  
    cout << "numar impar" ;
```

- instrucțiunea de selecție multiplă switch

switch (expresie)

```
{  
  case c1:  
    instructiune1;  
  [ break; ]  
  case c2:  
    instructiune2;  
  [ break; ]  
  ...  
  [ default:  
    instructiuni_d; ]  
}
```

expresie are valoare întreagă
break întrerupe execuția lui switch

- exemplu Problema rezolvată pag.103

- Temă p16/pag.103