

Aplicații cu structura decizională (alternativă)

dacă *expresie_logică*
atunci *secvența instrucțiunii1*
altfel *secvența instrucțiunii2*
sfârșit dacă

dacă *expresie_logică*
atunci *secvența instrucțiunii1*
sfârșit dacă

— În cazul că *expresie_selector*
Cazul *constantă1*:
secvența1
[iesire]
Cazul *constantă2*:
secvența2
[iesire]
...
Cazul *constantă_n*:
secvența_n
[iesire]
[altfel *secvența*]

- instrucțiunea if (dacă)

```
if (expresie_logică)
    i1 ;
[else
i2 ; ]
```

- instrucțiunea de selecție multiplă switch

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

- Problema rezolvata :

Se dau trei numere a, b, c.

Sa se afiseze dacă pot fi lungimile laturilor unui triunghi sau dacă nu pot fi.

Scrieti algoritmul si programul cu CodeBlocks.

date de intrare : a, b, c

date de iesire : mesajele "DA", "NU"

Indicatii : a, b, c trebuie să se verifice inegalitățile triunghiului :

$a, b, c > 0$

$a < b+c$

$b < a+c$

$c < a+b$

Algoritmul de rezolvare:

start

scrie "Introdu a: "

citeste a

scrie "Introdu b: "

citeste b

scrie "Introdu c: "

citeste c

daca $a > 0$ AND $b > 0$ AND $c > 0$

atunci

daca $a < b+c$

atunci

daca $b < a+c$

atunci

daca $c < a+b$

atunci

scrie "DA"

altfel

scrie "NU"

sfarsit daca

altfel

scrie "NU"

sfarsit daca

atfel

scrie "NU"

sfarsit daca

altfel

scrie "NU"

sfarsit daca

stop

Programul de scris cu Codeblocks :

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int a, b, c ;
    cout<<"Introdu a : " ;
    cin >> a ;
    cout<<"Introdu b : " ;
    cin >> b ;
    cout<<"Introdu c : " ;
    cin >> c ;
    if (a > 0 && b > 0 && c > 0 ) //1
        if (a < b+c) //2
            if (b < a+c) //3
                if (c < a+b) //4
                    cout << "DA";
                else //4
                    cout << "NU"; //4
            else //3
                cout << "NU"; //3
        else //2
            cout << "NU"; //2
    else //1
        cout << "NU" ; //1
    return 0;
}
```

De observat :

- imbricarea if-urilor si punctul de continuare dupa executarea unui if interior
- trebuie pusa perechea {} in ramura DA a if-urilor ?

Concluzii :

- inchiderea corecta a if-urilor este foarte importanta.
- Se vede mai usor prin scriere decalata.
- If-urile imbricate complica compunerea programelor. Este util sa folosim conditii compuse din mai multe propozitii cand este posibil

- **Tema** : Modificați algoritmul și programul de mai sus astfel încât :

- ✓ să fie o singură instrucțiune if cu o condiție compusă
- ✓ să calculați și afișați perimetrul și aria triunghiului folosind formulele :

Per = a+b+c

Aria = $\sqrt{p(p-a)(p-b)(p-c)}$ formula lui Heron p = semiperimetrul