Semelbauer Stefan

IF - STRUCTURA DE DECIZIE

Programele pe care le-am scris până acum fac lucruri destul de simple. De fapt fac aproape aceleași lucruri pe care le face un calculator de buzunar. Ar fi mult mai interesant dacă am putea executa anumite instrucțiuni doar dacă sunt îndeplinite niște condiții.

De exemplu Facebook ne arată notificări pentru mesaje în bara de sus doar atunci când primim mesaje noi.

```
Dacă avem mesaje necitite
    Atunci afișează o notificare
```

Un alt exemplu relevant sunt rețelele sociale. Dacă utilizatorul a dat click pe butonul de like, atunci numărul de like-uri trebuie să crească

```
Dacă cineva a dat click pe butonul de like
    Atunci crește numărul de like-uri pentru postare
```

În mod asemănător putem scrie instrucțiuni în C++ care să condiționeze comportamentul programului.

Exemplu

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

int main() {
    int a;
    cin>>a;
    if(a == 2) {
        cout<<"numarul citit este 2";
    }
    return 0;
}
```

Programul de mai sus va afișa un mesaj atunci când introducem numărul `2`.

```
#include <iostream>
using namespace std;
int main() {
    int a;
    cin>>a;
    if(a > 0) {
        cout<<"numarul citit este mai mare decât 0";
    }
    if(a <= 0) {
        cout<<"numarul citit nu este mai mare decât 0";
    }
    return 0;
```

```
}
```

Programul de mai sus afișează pe ecran dacă numărul introdus de către utilizator este mai mare decât 0 sau nu.

STRUCTURA INSTRUCTIUNII IF

if (condiție)

Așa cum am spus în lecția precedentă, instrucțiunea `if` ne permite să executăm instrucțiuni doar atunci când o anumită condiție este îndeplinită. Condiția este specificată între parantezele de după `if`.

Structura unei condiții

Condiția este o expresie care poate lua valoarea adevărat sau fals.

Exemple

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

int main() {
    int a, b;
    cin >> a >> b;
    if (a == b) {
        cout << "Numerele introduse sunt egale";
    }
    return 0;
}
```

Programul de mai sus verifică dacă două numere introduse sunt egale și afișează un mesaj pe ecran în caz afirmativ.

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

int main() {
    int a, b;
    cin >> a >> b;
    if (a != b) {
        cout << "Numerele introduse sunt diferite";
    }
    return 0;
}
```

În acest exemplu programul verifică dacă numere introduse sunt diferite.

Operatori

Pentru a scrie condițiile, limbajul C++ ne pune la dispoziție operatorii din tabelul de mai jos:

Operator	Explicație	Exemplu
`==`	Verifică dacă expresia din stânga este egală cu cea din dreapta	`if ((2+3)*5==25)`
`!=`	Verifică dacă expresia din stânga este diferită de cea din dreapta	`if (12+33!=(1+2)/3)`
`>`	Verifică dacă valoarea expresiei din stânga este mai mare decât valoarea expresiei din dreapta	`if (3+5>4)`
`<`	Verifică dacă valoarea expresiei din stânga este mai mică decât valoarea expresiei din dreapta	`if (3+5<14)`
`>=`	Verifică dacă valoarea expresiei din stânga este mai mare sau egală cu valoarea expresiei din dreapta	`if (3+5>=8)`
`<=`	Verifică dacă valoarea expresiei din stânga este mai mică sau egală cu valoarea expresiei din dreapta	`if (3+5<=14)`

Atenție!

Nu confunda operatorul `==` care testează dacă două expresii sunt egale cu `=` care stochează o valoare într-o variabilă.

{ și }

Așa cum acoladele lui `int main()` conțin în interiorul lor instrucțiunile programului scris de către noi, la fel acoladele unui `if` conțin în interiorul lor instrucțiunile care trebuie executate doar dacă condiția din interiorul lui `if` este adevărată.

CUTII

După ce a devenit foarte popular, Mark a primit în total A cadouri de la fanii săi. Acum vrea să se mute într-o casă mai mare și trebuie să pună toate cadourile în cutii pentru a se putea muta. Pentru că e ordonat, Mark vrea să pună în fiecare cutie **exact** B cadouri. Putem să ne imaginăm că Mark poate să facă rost de oricâte cutii oricât de mari.

Spuneți dacă Mark poate să își mute cadourile respectând condiția din enunț.

Date de intrare

Se citesc de la tastatură cele două numere A și B.

Date de ieșire

Programul va afișa pe ecran posibil dacă e posibil ca Mark să se mute respectând condiția sau imposibil în caz contrar.

Restricții și precizări

Numerele sunt strict pozitive și se încadrează în tipul de date int.

Exemplu

Date de intrare	Date de ieșire
6 3	posibil
6 4	imposibil
5 3	imposibil
25 5	posibil

Explicație

Pentru primul exemplu avem 6 cadouri și trebuie să punem exact 3 cadouri în fiecare cutie. Putem rezolva acest lucru folosind 2 cutii.

În al doilea exemplu avem 6 cadouri și trebuie să punem exact 4 cadouri în fiecare cutie. Putem pune în prima cutie 4 cadouri și pe urmă vom rămâne cu 2 cadouri pe care nu le putem pune într-o cutie deoarece trebuie să avem exact 4 cadouri în fiecare dintre cutii, deci răspunsul e imposibil.

CODEBLOCKS ONLINE (TESTER-ul lor).

REZOLVAREA MEA

PITAGORA

Cerință

Se dau 3 numere naturale, a, b și c, care respectă condiția a < b < c. Sarcina ta este să determini dacă tripletul (a, b, c) respectă următoarea condiţie: $a^2+b^2=c^2$.

Date de intrare

De la tastatură se vor citi cele trei numere naturale a,b,c.

Date de ieşire

Pe ecran se va afişa DA, în cazul în care cele trei numere respectă condiţia dată, sau NU, în caz contrar.

Restricţii şi precizări

$1 \leq a < b < c \leq 1000$

Exemplu

Date de intrare	Date de ieşire
3 4 5	DA
2 5 7	NU

Mai ții minte teorema lui Pitagora?

Din Teorema lui Pitagora reiese că în orice triunghi dreptunghic, suma pătratelor lungimilor laturilor alăturate unghiului drept (catetelor) e egală cu pătratul lungimii laturii opuse (ipotenuzei). ($cateta1^2$ + $cateta2^2$ = $ipotenuza^2$)

În problema noastră, vrem să verificăm dacă a, b și c sunt laturile unui triunghi dreptunghic, a și b fiind catetele, iar c ipotenuza.

Teorema lui Pitagora e folosită foarte frecvent și pentru a afla distanța dintre 2 puncte în plan, lucru care este folosit des în jocurile video alături de alte concepte din geometrie.

CODEBLOCKS ONLINE (TESTER-ul lor).

REZOLVAREA MEA

CONDITII COMPUSE

Câteodată nu este de ajuns o singură condiție pentru a realiza ce ne dorim. Alteori avem nevoie de instrucțiuni mai complicate.

De exemplu Facebook ne arată notificări și dacă avem un comentariu nou la o postare, dar și atunci când primim like la postare.

```
Dacă utilizatorul are un like nou sau dacă are un comentariu nou
    Afișează o notificare
```

Acest lucru se poate realiza în C++ în felul următor:

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

int main() {
    int a;
    cin>>a;
    if (a > 2 && a <= 10) {
        cout<<"Numarul introdus este mai mare decat 2 si mai mic sau egal decat 10";
    }
    return 0;
}
```

Programul de mai sus afișează un mesaj dacă numărul introdus este mai mare decât 2 și mai mic sau egal cu 10. Pentru a realiza acest lucru am folosit operatorul `&&` (și) pentru a lega două condiții într-o nouă condiție care să fie verificată de către `if`.

Operatorul `&&`

Denumire: și

Exemplu: `if (a < 9 && a > 1)`

Explicație: Condiția este adevărată dacă condiția din stânga **și** cea din dreapta sunt ambele adevărate.

Operatorul `||`

Denumire: sau

Exemplu: `if (a % 2 == 0 || a % 3 == 0)`

Explicație: Condiția este adevărată dacă condiția din stânga **sau** cea din dreapta este adevărată.

Înlănțuirea condițiilor

Acești doi operatori ne permit să înlănțuim oricâte condiții în interiorul unui if. Pentru a permite condiții mai complexe, putem folosi și paranteze.

Exemplu

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

int main() {
    int a, b;
    cin>>a>>b;
    if ((a < 0 && b < 0) || (a > 0 && b > 0)) {
        cout<<"Numerele introduse au același semn";
    }
    return 0;
}
```

Programul de mai sus verifică dacă ambele numere citite sunt ambele negative sau ambele pozitive.

Atenție Așa cum în cazul operațiilor aritmetice există o ordine (`2+3*4` este egal cu 14 deoarece înmulțirea se face înaintea adunării), la fel și în cazul condițiilor, && se face înaintea lui ||, deci condiția de mai sus poate fi rescrisă fără paranteze în felul următor: `if (a < 0 && b < 0 || a > 0 && b > 0)`

FIZZBUZZ

Se dă un număr pozitiv N. Să se afișeze fizz dacă acesta este divizibil cu 2, buzz, dacă este divizibil cu 3 sau fizzbuzz, dacă este divizibil și cu 2 și cu 3.

Date de intrare

Se citește la tastatură numărul N.

Date de ieșire

Programul va afișa pe ecran cuvintele cerute.

Restricții

0 < N < 10000

Exemplu

Date de intrare	Date de ieșire
124	fizz
15	buzz

CODEBLOCKS ONLINE (TESTER-ul lor).

REZOLVAREA MEA

IF INLANTUIT

Așa cum am spus în prima lecție, codul din interiorul acoladelor `{ ... }` unui `if` se va executa doar dacă condiția din `if` este îndeplinită.

Asta ne permite să scriem programe mai complicate precum următorul:

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

int main() {
    int a,b;
    cin>>a>>b;
    if (a>=0) {
        if (b>=0) {
            cout<<"Ambele numere citite sunt pozitive";
        }
        if (b<0) {
            cout<<"Primul numar citit este pozitiv, iar al doilea negativ";
        }
    }
    if (a<0) {
        if (b<0) {
            cout<<"Ambele numere sunt negative";
        }
        if(b>=0) {
            cout<<"Primul numar citit este negativ, iar al doilea pozitiv";
        }
    }
    return 0;
}
```

De exemplu dacă rulăm programul de mai sus, în funcție de numerele introduse de către tine, vor fi afișate diverse valori. Dacă prima valoare introdusă ar fi `2` și a doua `-1`, programul va afișa pe ecran `Primul numar citit este pozitiv, iar al doilea negativ` deoarece condiția primului `if` va fi adevărată, a celui de-al doilea falsă și a celui de-al treilea adevărată. Pentru a înțelege mai bine, rulează programul pe mai multe exemple pentru a vedea ce se întâmplă.

În practică de multe ori avem nevoie să scriem programe foarte complexe. Un exemplu foarte bun este software-ul folosit de către avioane pentru a zbura. Piloții de avion aproximează că aproximativ 90% dintr-un zbor normal este coordonat de autopilotul avionului.

Un astfel de soft are de luat multe decizii în funcție de condițiile atmosferice, celelalte avioane apropiate și distanța față de aeroportul destinație. În cazul autopilotului, viețile pasagerilor depind de cât de bun este codul sursă scris de către programatori.

ELSE

Câteodată când condiția unui if nu este îndeplinită, vrem să executăm un set de instrucțiuni. De exemplu dacă dorim să afișăm dacă un număr este par sau impar, am face în felul următor:

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

int main() {
    int a;
    cin>>a;
    if (a % 2 == 0) {
        cout<<"Numarul citit este par";
    }
    if (a % 2 != 0) {
        cout<<"Numarul citit este impar";
    }
    return 0;
}
```

Totuși este deranjant să scriem din nou condiția opusă condiției din primul if de fiecare dată când dorim ca programul să se comporte în acest fel.

Pentru a executa o serie de instrucțiuni atunci când condiția din primul if nu este îndeplinită putem folosi else { ... }. Instrucțiunile din interiorul acoladelor lui else se vor executa doar atunci când condiția din if nu este îndeplinită.

Exemplu

Codul de mai sus poate fi rescris în felul următor cu ajutorul lui else

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

int main() {
    int a;
    cin>>a;
    if (a % 2 == 0) {
        cout<<"Numarul citit este par";
    } else {
        cout<<"Numarul citit este impar";
    }
    return 0;
}
```

Atenție! else trebuie pus totdeauna după un if de care va aparține.

MAXIM

Se dau două numere A și B. Se cere să se afișeze cel mai mare dintre ele.

Date de intrare

Se citesc de la tastatură două numere, reprezentând valorile A, respectiv B.

Date de ieșire

Programul va afișa pe ecran maximul celor 2 numere.

Restricții

Numerele se încadrează in tipul de date int.

Exemplu

Date de intrare	Date de ieșire
123 256	256

CODEBLOCKS ONLINE (TESTER-ul lor).

REZOLVAREA MEA

CRESCATOR 2

Se dau două numere întregi `A` și `B`. Să se afișeze în ordine crescătoare.

Date de intrare

Se citesc de la tastatură cele două numere.

Date de ieșire

Programul va afișa pe ecran numerele ordonate crescător, separate prin spații.

Restricții și precizări

Numerele se încadrează în tipul de date `int`.

Exemplu

Date de intrare	Date de ieșire
`56 -99`	`-99 56`

CODEBLOCKS ONLINE (TESTER-ul lor).

REZOLVAREA MEA

ELSE IF

Deși `else` ne ajută să scriem mai puțin, câteodată avem de făcut lucruri mai complicate.

Să presupunem că dorim să afișăm dacă un număr se divide cu `2` sau dacă se divide cu `3`, dar nu se divide cu `2` sau dacă se divide cu `5`, dar nu se divide cu `2` și cu `3`. O primă soluție ar fi următoarea:

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

int main() {
    int a;
    cin>>a;
    if (a % 2 == 0) {
        cout<<"Numarul citit se divide cu 2";
    }
    if (a % 2 != 0 && a % 3 == 0)  {
        cout<<"Numarul citit se divide cu 3, dar nu cu 2";
    }
    if (a % 2 != 0 && a % 3 != 0 && a % 5 == 0) {
        cout<<"Numarul citit se divide cu 5, dar nu cu 2 si 3";
    }
    return 0;
}
```

Dacă am folosi `else` programul ar fi scurtat puțin, dar ar fi mai greu de înțeles.

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

int main() {
    int a;
    cin>>a;
    if (a % 2 == 0 ) {
        cout<<"Numarul citit se divide cu 2";
    } else {
        if (a % 3 == 0)  {
            cout<<"Numarul citit se divide cu 3, dar nu cu 2";
        } else {
            if (a % 5 == 0) {
                cout<<"Numarul citit se divide cu 5, dar nu cu 2 si 3";
            }
        }
    }
    return 0;
}
```

Pentru acest gen de cazuri este potrivită instrucțiunea `else if`:

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

int main() {
    int a;
    cin>>a;
    if (a % 2 == 0 ) {
        cout<<"Numarul citit se divide cu 2";
    } else if (a % 3 == 0)  {
        cout<<"Numarul citit se divide cu 3, dar nu cu 2";
    } else if (a % 5 == 0) {
        cout<<"Numarul citit se divide cu 5, dar nu cu 2 si 3";
    } else {
        cout<<"Numarul nu se divide nici cu 2, nici cu 3 si nici cu 5";
    }
    return 0;
}
```

Deși nu făcea parte din cerința problemei, am adăugat ultimul `else` pentru a ilustra mai bine modul de funcționare. Când avem mai multe `else if`, maxim o condiție va fi îndeplinită. Condiția dintr-un `else if` este verificată doar după ce condiția din `else if`-ul anterior a fost verificată și nu a fost îndeplinită.

Exemplu

Să se verifice dacă diferența a două numere este negativă sau este egală cu 0.

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

int main() {
    int a,b;
    cin>>a>>b;
    if (a - b < 0) {
        cout<<"Diferenta este negativa";
    } else if (a - b == 0) {
        cout<<"Diferenta este egala cu 0";
    }
    return 0;
}
```

CRESCATOR 3

Se dau `3` numere întregi `A`, `B` și `C`, nu neapărat distincte. Să se afișeze în ordine crescatoare.

Date de intrare

Se citesc de la tastatură cele trei numere.

Date de ieșire

Programul va afișa pe ecran numerele ordonate crescător, separate prin spații.

Restricții

Numerele se încadrează in tipul de date `int`.

Exemplu

Date de intrare	Date de ieșire
`3 -1 20`	`-1 3 20`

CODEBLOCKS ONLINE (TESTER-ul lor).

REZOLVAREA MEA

FINAL MODUL - CE URMEAZA?

Dacă vrei să te pregătești personal cu mine și Bianca, completează formularul dând click aici și înscrie-te în programul nostru de mentorat în programare.

De abia aștept să începem să facem treabă,

Petru