

OPERATORI

Un program în care putem face doar citiri, declarații și atribuiri nu are cum să fie prea util. De multe ori avem nevoie să folosim operații aritmetice în programele pe care le scriem. De exemplu algoritmul Page Rank folosit de Google pentru a decide ordinea în care paginile sunt afișate pentru o căutare folosește operații aritmetice.

Mai jos este un tabel cu operatorii aritmetici pe care îi putem folosi în C++.

Operație	Operator în C++	Explicație	Exemplu
Adunare	+	Adună cele două numere	2+3 rezultatul va fi 5
Scădere	-	Scade al doilea număr din primul	2-3 rezultatul va fi -1
Înmulțire	*	Înmulțește cele două numere	5*7 rezultatul va fi 35
Împărțire	/	împarte primul număr la cel de-al doilea. În cazul în care lucrăm cu numere întregi, rezultatul împărțirii va fi aproximat prin lipsă.	2/3 rezultatul va fi 0, iar pentru 14/5 va fi 2
Restul împărțirii	%	Află restul împărțirii primului număr la cel de-al doilea.	14%5 rezultatul va fi 4

Atenție!

Deoarece variabilele `int` nu pot stoca numere raționale, atunci când folosim împărțirea și numerele nu se împart exact unul la altul, rezultatul va fi aproximat prin lipsă. De exemplu `14 / 3` va fi egal cu `4`.

Exemple

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

int main() {
    int a=3, b=5, rez;
    rez=a-b;
    cout<<rez;
    return 0;
}
```

Programul face diferența dintre variabila `a` și variabila `b` și o pune în variabila `rez`. Se va afișa `-2`.

```
#include <iostream>
```

```
using namespace std;

int main() {
    int x=4, y=6, calcul;
    calcul=x*x-2*x*y+y*y+5;
    cout<<calcul;
    return 0;
}
```

În cadrul operațiilor putem folosi oricâte numere, variabile și operații. Pe ecran se va afișa 9, adică rezultatul calculului $(4*4)-(2*4*6)+(6*6)+5$

Paranteze

În cadrul operațiilor putem folosi și paranteze. Spre deosebire de notația matematică, în C++ vom folosi doar paranteze rotunde (și). Putem folosi oricâte paranteze dorim cu condiția ca orice paranteză deschisă să fie și închisă. De exemplu expresia aritmetică $(4+(56*(42-((23+12)-3)/2)\%6))*88$ este corectă.

Exemplu

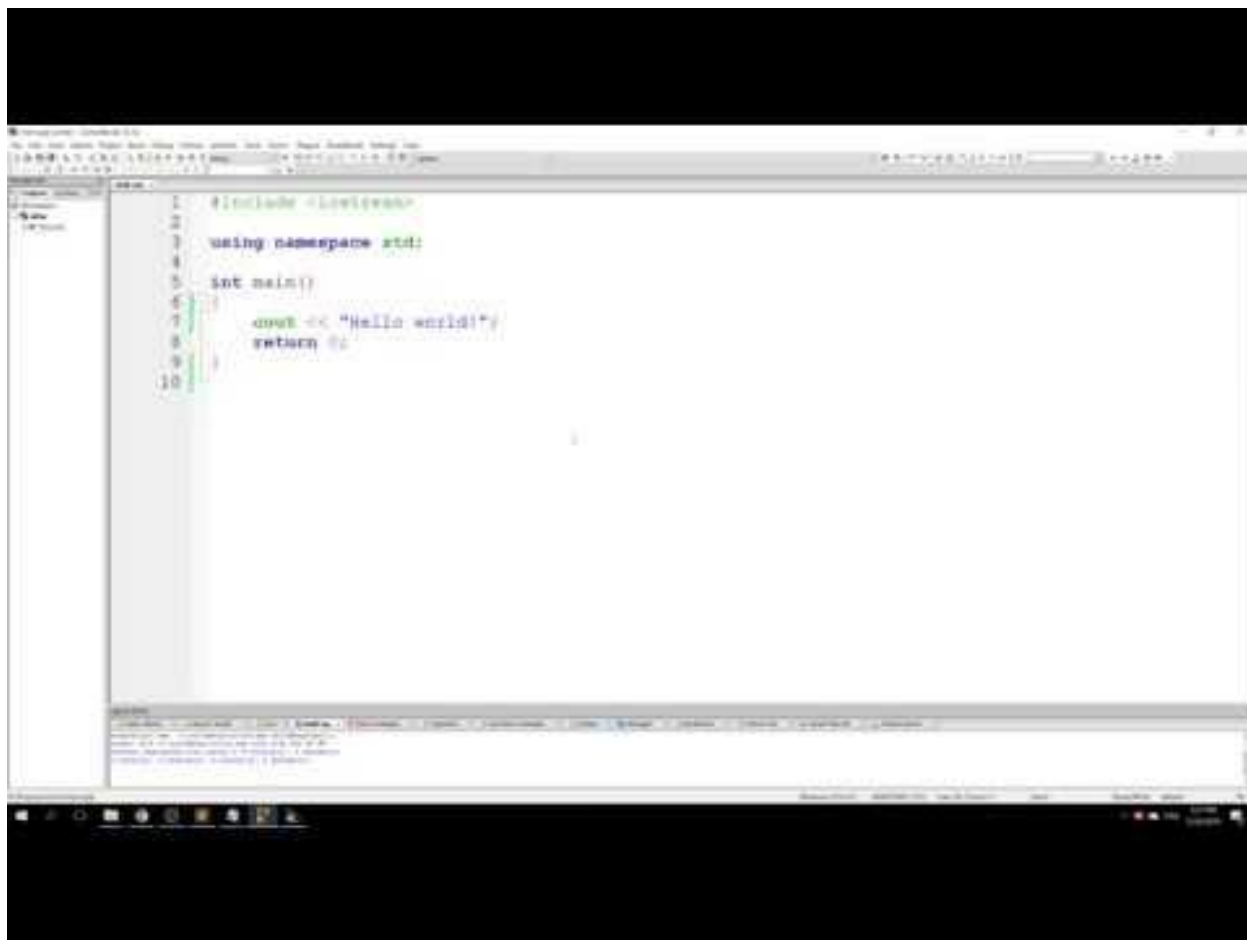
```
#include <iostream>
using namespace std;

int main() {
    int x=4, y=6, calcul;
    calcul=(x+y)*(x+y);
    cout<<calcul;
    return 0;
}
```

În acest caz se va afișa 100.

ÎNAINTE SĂ REZOLVI PRIMA PROBLEMA

Înainte să începi să rezolvi prima problemă mai grea de pe platformă, urmărește până la capăt cu atenție tutorialul de mai jos pentru a înțelege exact cum funcționează citirea și afișarea, dar și pentru a vedea cum vor fi evaluate programele



tale pe platformă:

ADUNARE

În această problemă va trebui să citești 2 numere întregi de la tastatură și să afișezi suma lor pe ecran.

Date de intrare

Pe prima linie se vor afla două numere întregi a și b separate printr-un spațiu.

Asta înseamnă că atunci când vrei să testezi programul, vei introduce ambele numere cu un spațiu între ele.

Date de ieșire

Pe ecran se va afișa:

- Suma celor doua numere;

Restricții

- Suma numerelor va fi mai mică decât 2 000 000 000
- Suma numerelor va fi mai mare decât -2 000 000 000

Exemplu

Date de intrare	Date de ieșire
1 2	3

CODEBLOCKS ONLINE (TESTER-ul lor).

[REZOLVAREA MEA](#)

CALCUL

Se dau două numere întregi a și b . Se cere să se calculeze valoarea expresiei $(a-b)^3$.

Date de intrare

Se citesc de la tastatură două numere întregi, reprezentând valorile a , respectiv b .

Date de ieșire

Programul va afișa pe ecran valoarea expresiei cerute.

Restricții

$|a|, |b| \leq 100$ $|a|$ reprezintă modulul numărului a .

Exemplu

Date de intrare	Date de ieșire
5 1	64

CODEBLOCKS ONLINE (TESTER-ul lor).

[REZOLVAREA MEA](#)

OPERATORI – II

Așa cum vom vedea în lecțiile următoare, de multe ori va trebui să creștem o variabilă cu o valoare sau să facem modificări simple asupra unei variabile. În mod normal dacă dorim să creștem variabila `a` cu `4` putem face `a = a + 4;`. În C++ putem scrie aceeași expresie într-un mod mai scurt: `a += 4`. Putem face acest lucru cu toate operațiile:

Operator	Expresie echivalentă	Explicație
<code>a += b</code>	<code>a = a + b</code>	Adună valoarea lui <code>b</code> la <code>a</code>
<code>a -= b</code>	<code>a = a - b</code>	Scade din <code>a</code> valoarea lui <code>b</code>
<code>a *= b</code>	<code>a = a * b</code>	Înmulțește valoarea lui <code>a</code> cu <code>b</code>
<code>a /= b</code>	<code>a = a / b</code>	Îl împarte pe <code>a</code> la <code>b</code>
<code>++a</code>	<code>a = a + 1</code>	Crește valoarea lui <code>a</code> cu <code>1</code>
<code>--a</code>	<code>a = a - 1</code>	Scade valoarea lui <code>a</code> cu <code>1</code>

Exemplu

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

int main() {
    int a=3,b=5;
    ++a;
    b*=a;
    cout<<a<<" "<<b;
    return 0;
}
```

Programul de mai sus va afișa `4 20`.

LIKE

Mark este un băiat foarte popular, cu foarte mulți prieteni pe Facebook. Ca să își afle popularitatea de pe rețeaua socială, Mark vrea să știe câte like-uri are în total la toate cele n postări ale lui, știind că la fiecare postare are exact x like-uri.

Date de intrare

Se vor citi de la tastatură n și x , cu semnificațiile din enunț.

Date de ieșire

Pe ecran se va afișa un singur număr, numărul total de like-uri pe care le-a primit Mark.

Restricții

- $0 \leq n < 1001$
- $0 \leq x < 1001$

Exemplu

Date de intrare	Date de ieșire
10 5	50

CODEBLOCKS ONLINE (TESTER-ul lor).

[REZOLVAREA MEA](#)