

Booleovu algebru susrećemo na svakom koraku u našim programima. Svaki puta kada napišemo IF naredbu, kao uvjet definiramo **izraz booleove algebre**. U mnogim programima, potencijalne greške skrivaju se upravo iza tih booleovih izraza... Zato je kod svakog pravog ispitivanja potrebno uzeti u obzir skoro pa svaku kombinaciju bitova (boolova) za zadani izraz.

Kao primjer prikažimo sljedeći IF uvjet:

```
( a==1 OR b==1 AND c==1 )
```

$(a==1)$ je tzv. **literal**, odnosno izraz koja se prevodi u bitovnu vrijednost 0 ili 1, pa taj literal označimo s A. Slično vrijedi i za $(b==1) \rightarrow B$ te $(c==1) \rightarrow C$.

Sada izraz pojednostavimo ovakvim zapisom:

A OR B AND C . Prevedimo sad zapis u oblik gdje OR zamijenimo s +, a AND s *.
A + B * C . Dobili smo pojednostavljeni zapis izraza booleove algebre!

Isti možemo riječima opisati ovako: Izraz je istinit ako je A istinit ili ako su B i C istovremeno istiniti, a to se može dogoditi u ukupno **5** od mogućih 8 kombinacija vrijednosti literala A, B i C. Napišite program koji će za zadani izraz booleove algebre ispisati koliko ukupno kombinacija vrijednosti literala dovodi vrijednost izraza u istinu.

Ulazni podaci:

- Izraz booleove algebre s literalima (mala slova eng. abecede) i operatorima AND (*) i OR (+). Operacija AND uvijek ima veći prioritet od operacije OR. Ukupan ulazni niz može biti duljine najviše 19 znakova (10 literala i 9 operacija). Razmaci se neće pojavljivati.

Izlazni podaci:

- Broj kombinacija vrijednosti literala u kojima zadani booleov izraz vraća istinu

Sustavna ograničenja:

- 500 milisekundi za izvršavanje po test primjeru
- 32 MB HEAP memorija, 32 MB STACK memorija

Test primjeri:

ULAZ	a+b	a+b+c	a*b+c*d+d*e
IZLAZ	3	7	17

Objašnjenje 2. test primjera: izraz vraća istinu za sve kombinacije osim $a = b = c = 0$.

Naziv datoteke: **bool.c** ili **bool.cpp**

Također su dozvoljene i ekstenzije za BASIC (.bas) i Pascal (.pas) ukoliko radite u jednom od tih jezika.