

Vaje pri predmetu Programiranje 2

Teden 1: Krmilni stavki

Vsota po modulu

Naloga

Napišite program, ki prebere števili a in b in izpiše njuno vsoto po modulu 10.

Vhod

Na vhodu je najprej zapisano celo število $a \in [0, 9]$, nato sledi presledek, tretji znak pa je celo število $b \in [0, 9]$.

Izhod

Izpišite iskani rezultat.

Testni primer

Vhod:

9 7

Izhod:

6

Vsa enaka

Naloga

Napišite program, ki prebere število n in še n števil, nato pa izpiše 1, če so vsa izmed teh n števil med seboj enaka, in 0, če to ne drži.

Vhod

Na vhodu je najprej zapisano celo število $n \in [0, 9]$, nato pa sledi presledek in n celih števil z intervala $[0, 9]$, ločenih s presledkom.

Izhod

Izpišite 1 oziroma 0.

Testni primer 1

Vhod:

7 3 3 4 3 4 3 3

Izhod:

0

Testni primer 2

Vhod:

4 7 7 7 7

Izhod:

1

Drugo največje

Naloga

Napišite program, ki prebere število n in še n števil, nato pa izpiše drugo največje med temi n števili. Če je več števil največjih, je drugo največje število kar enako največjemu.

Vhod

Na vhodu je najprej zapisano celo število $n \in [2, 9]$, nato pa sledi presledek in n celih števil z intervala $[0, 9]$, ločenih s presledkom.

Izhod

Izpišite iskano število.

Testni primer 1

Vhod:

7 5 2 3 8 4 6 3

Izhod:

6

Testni primer 2

Vhod:

4 7 5 5 7

Izhod:

7

Vaje pri predmetu Programiranje 2

Teden 2: Krmilni stavki

Pitagorejske trojice

Naloga

Napišite program, ki prebere pozitivni celi števili m in n in izpiše, koliko celih števil $c \in [m, n]$ lahko zapišemo v obliki $a^2 + b^2 = c^2$, kjer sta a in b pozitivni celi števili. (Trojico (a, b, c) s takšno lastnostjo imenujemo *pitagorejska trojica*.)

Vhod

Na vhodu sta zapisani celi števili $m \in [1, 10^4]$ in $n \in [m, 10^4]$, ločeni s presledkom.

Izhod

Izpišite število iskanih števil.

Testni primer 1

Vhod:

10 20

Izhod:

5

V tem primeru imamo 5 števil z iskano lastnostjo: 10 ($10^2 = 6^2 + 8^2$), 13 ($13^2 = 5^2 + 12^2$), 15 ($15^2 = 9^2 + 12^2$), 17 ($17^2 = 8^2 + 15^2$) in 20 ($20^2 = 12^2 + 16^2$).

Optimalno ugibanje

Naloga

Mirko si zamisli celo število na intervalu $[a, b]$, računalnik pa ga uganjuje po optimalnem postopku, torej tako, da vsakokrat poskuša z navzdol zaokroženim povprečjem med spodnjo in zgornjo mejo trenutnega intervala. Na vsak računalnikov poskus se Mirko odzove tako, da vtipka 1 (če je računalnikov poskus manjši od izbranega števila) ali -1 (če je računalnikov poskus prevelik). Mirko se lahko igre predčasno naveliča, lahko pa se tudi zgodi, da računalniku postreže s protislovnimi odgovori.

Napišite program, ki prebere števili a in b ter zaporedje Mirkovih odgovorov računalniku (to se vedno zaključi s številom 0) in izpiše eno od sledečega:

- izbrano število, če ga je na podlagi Mirkovih odgovorov mogoče nedvoumno določiti;
- spodnjo in zgornjo mejo najožjega intervala, na katerem se glede na Mirkove odgovore nahaja izbrano število;
- niz PROTISLOVJE, če so Mirkovi odgovori v medsebojnem protislovju.

Vhod

V prvi vrstici vhoda sta podani celi števili $a \in [0, 10^9]$ in $b \in [a, 10^9]$, ločeni s presledkom. Sledi zaporedje vrstic s številom 1 ali -1 . Zadnja vrstica vsebuje število 0.

Izhod

Izpišite izbrano število (prvi scenarij), spodnjo in zgornjo mejo najožjega intervala, ki vsebuje izbrano število (drugi scenarij), oziroma niz PROTISLOVJE (tretji scenarij). Meji intervala naj bosta ločeni s presledkom.

Testni primer 1

Vhod:

```
10 30
1
-1
-1
0
```

Izhod:

```
21
```

Računalnik najprej poskuša s številom 20 ($= \lfloor (10 + 30) / 2 \rfloor$), nato s 25 ($= \lfloor (21 + 30) / 2 \rfloor$), nato z 22 ($= \lfloor (21 + 24) / 2 \rfloor$), po tretjem odgovoru pa ve, da si je Mirko izbral število 21.

Testni primer 2

Vhod:

```
10 30
1
-1
0
```

Izhod:

```
21 24
```

Testni primer 3

Vhod:

```
10 30
-1
-1
-1
1
1
-1
0
```

Izhod:

PROTISLOVJE

Vaje pri predmetu Programiranje 2

Teden 3: Funkcije

Prijateljska števila

Naloga

Pozitivni celi števili sta *prijateljski*, če sta medsebojno različni in če je vsota njunih pravih deliteljev (tj. deliteljev brez števila samega) navzkrižno enaka drugemu številu. Na primer, števili 220 in 284 sta prijateljski, ker je vsota pravih deliteljev števila 220 ($1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110$) enaka 284, vsota pravih deliteljev števila 284 ($1 + 2 + 4 + 71 + 142$) pa znaša 220.

Napišite program, ki za podano število n preveri, ali ima prijateljsko število. Če ga ima, naj ga izpiše, sicer pa naj izpiše niz NIMA.

Vhod

Na vhodu je zapisano celo število $n \in [1, 10^6]$.

Izhod

Izpišite prijateljsko število števila n oziroma niz NIMA, če n nima prijateljskega števila.

Testni primer 1

Vhod:

30

Izhod:

NIMA

Testni primer 2

Vhod:

220

Izhod:

284

Lychrelova števila

Naloga

Pozitivnemu celemu številu prištejmo njegovo zrcalno sliko in postopek ponavljajmo, dokler ne dobimo palindroma (števila, ki se z obeh strani bere enako). Na primer, če pričnemo s številom 64, potrebujemo dve iteraciji: po prvi dobimo število 110 ($64 + 46$), po drugi pa palindromno število 121 ($110 + 11$). Nekatera števila zahtevajo več iteracij:

- $77 \rightarrow 154 \rightarrow 605 \rightarrow 1111$ (3 iteracije)
- $182 \rightarrow 463 \rightarrow 827 \rightarrow 1555 \rightarrow 7106 \rightarrow 13123 \rightarrow 45254$ (6 iteracij)

Števila, ki nas po opisanem postopku nikoli ne pripeljejo do palindroma, se imenujejo *Lychrelova števila*. Ni dokazano, ali kakšno tovrstno število sploh obstaja, najmanjši kandidat zanj pa je število 196.

Označimo z $f_i(n)$ število, ki ga iz števila n dobimo po i iteracijah opisanega postopka. V tej nalogi bomo za *domnevno Lychrelovo število* proglasili vsako število n , pri katerem nobeno od števil $f_1(n), f_2(n), \dots, f_k(n)$ ni palindrom. (Palindromskost samega števila n nas ne zanima.) Poleg tega bomo postopek predčasno prekinili, takoj ko bomo naleteli na število $f_i(n) > 10^{17}$. Prvo število nad 10^{17} bomo pri preverjanju palindromskosti še upoštevali, naslednjega pa ne več.

Napišite program, ki prebere število k ter spodnjo in zgornjo mejo intervala, izpiše pa število domnevnih Lychrelvih števil na podanem intervalu.

Vhod

Vhod je sestavljen iz celih števil $k \in [1, 100]$, $a \in [1, 10^4]$ in $b \in [a, 10^4]$, ločenih s presledkom.

Izhod

Izpišite število domnevnih Lychrelvih števil na intervalu $[a, b]$.

Testni primer 3

Vhod:

```
100 1 1000
```

Izhod:

```
13
```

Števila z iskano lastnostjo so v tem primeru 196, 295, 394, 493, 592, 689, 691, 788, 790, 879, 887, 978 in 986.

Vaje pri predmetu Programiranje 2

Teden 4: Tabele

Permutacija

Naloga

Napišite program, ki prebere število n in zaporedje n števil in izpiše **DA**, če je prebrano zaporedje permutacija množice $\{0, \dots, n-1\}$, in **NE**, če to ne drži.

Vhod

V prvi vrstici je podano celo število $n \in [1, 10^6]$, v drugi pa zaporedje n celih števil z intervala $[-10^9, 10^9]$, ločenih s po enim presledkom.

Izhod

Izpišite **DA** oziroma **NE**.

Testni primer 1

Vhod:

```
5
2 3 0 4 1
```

Izhod:

```
DA
```

Testni primer 2

Vhod:

```
3
2 3 1
```

Izhod:

```
NE
```

Največje podzaporedje

Naloga

Napišite program, ki prebere število n in zaporedje n števil in izpiše vsoto členov nepraznega strnjenega podzaporedja, ki ima največjo vsoto.

Vhod

V prvi vrstici je podano celo število $n \in [1, 10^6]$, v drugi pa je nanizanih n celih števil z intervala $[-1000, 1000]$, ločenih s po enim presledkom.

Izhod

Izpišite iskano vsoto.

Testni primer

Vhod:

```
8  
-3 -7 5 9 -6 7 -4 1 2
```

Izhod:

```
15
```

Podzaporedje z največjo vsoto je $\langle 5, 9, -6, 7 \rangle$.

Vaje pri predmetu Programiranje 2

Teden 5: Kazalci, tabele, nizi

Naloga

V datoteki `naloga.c` napišite sledeče funkcije:

- `int vsota(int* zac, int* kon)`

Kazalca `zac` in `kon` kažeta na elementa iste tabele. Element, na katerega kaže kazalec `kon`, se *ne* nahaja pred elementom, na katerega kaže kazalec `zac`. Funkcija naj vrne vsoto vseh elementov od vključno elementa, na katerega kaže kazalec `zac`, do vključno elementa, na katerega kaže kazalec `kon`.

- `void indeksInKazalec(int* t, int* indeks, int** kazalec)`

Kazalec `t` kaže na začetek tabele. Če ima spremenljivka, na katero kaže kazalec `indeks`, vrednost -1 , naj funkcija v to spremenljivko vpiše indeks elementa v tabeli, na katerega kaže kazalec, na katerega kaže kazalec `kazalec`. V nasprotnem primeru pa naj funkcija v spremenljivko, na katero kaže kazalec `kazalec`, vpiše kazalec na element, na čigar indeks kaže kazalec `indeks`.

- `void frekvenceCrk(char* niz, int** frekvence)`

Funkcija naj v spremenljivko, na katero kaže kazalec `frekvence`, vpiše kazalec na začetek novoustvarjene tabele frekvenc posameznih črk v nizu; i -ti element tabele naj vsebuje število pojavitev i -te črke v angleški abecedi. Funkcija naj ne razlikuje med velikimi in malimi črkami. Na primer, niz `baNanA` vsebuje tri črke `A`.

Vaje pri predmetu Programiranje 2

Teden 6: Rekurzija

Vsote I

Naloga

Napišite program, ki prebere števili n in k in izpiše, na koliko načinov lahko število n zapišemo kot vsoto števil med 1 in vključno k . Na primer, pri $n = 7$ in $k = 5$ imamo 13 načinov:

$$\begin{aligned}7 &= 5 + 2 \\7 &= 5 + 1 + 1 \\7 &= 4 + 3 \\7 &= 4 + 2 + 1 \\7 &= 4 + 1 + 1 + 1 \\7 &= 3 + 3 + 1 \\7 &= 3 + 2 + 2 \\7 &= 3 + 2 + 1 + 1 \\7 &= 3 + 1 + 1 + 1 + 1 \\7 &= 2 + 2 + 2 + 1 \\7 &= 2 + 2 + 1 + 1 + 1 \\7 &= 2 + 1 + 1 + 1 + 1 + 1 \\7 &= 1 + 1 + 1 + 1 + 1 + 1 + 1\end{aligned}$$

Vhod

Na vhodu sta zapisani celi števili $n \in [1, 400]$ in $k \in [1, n]$, ločeni s presledkom.

Izhod

Izpišite število načinov. Rezultat ne bo večji od $2^{63} - 1$.

Testni primer 1

Vhod:

7 5

Izhod:

13

Vsote II

Naloga

Napišite program, ki prebere števili n in k in izpiše vsa padajoča zaporedja števil med 1 in vključno k , katerih vsota znaša n .

Vhod

Na vhodu sta zapisani celi števili $n \in [1, 30]$ in $k \in [1, n]$, ločeni s presledkom.

Izhod

Zaporedja izpišite v padajočem leksikografskem vrstnem redu: zaporedje $\langle a_1, a_2, \dots, a_s \rangle$ naj se izpiše pred zaporedjem $\langle b_1, b_2, \dots, b_s \rangle$ natanko v primeru, če obstaja tak $i \geq 1$, da velja $a_i > b_i$, za vse $j < i$ pa velja $a_j = b_j$.

Vsako zaporedje izpišite v svoji vrstici, členi zaporedja pa naj bodo med seboj ločeni z zaporedjem presledka, znaka + in presledka.

Testni primer 1

Vhod:

7 5

Izhod:

```
5 + 2
5 + 1 + 1
4 + 3
4 + 2 + 1
4 + 1 + 1 + 1
3 + 3 + 1
3 + 2 + 2
3 + 2 + 1 + 1
3 + 1 + 1 + 1 + 1
2 + 2 + 2 + 1
2 + 2 + 1 + 1 + 1
2 + 1 + 1 + 1 + 1 + 1
1 + 1 + 1 + 1 + 1 + 1 + 1
```

Vaje pri predmetu Programiranje 2

Teden 7: Rekurzija 2

Problem nahrbtnika 1

Naloga

Nahrbtnik bi radi napolnili s predmeti s čimvečjo skupno ceno, pri tem pa smo seveda omejeni z razpoložljivo prostornino. Napišite program, ki prebere prostornino nahrbtnika, število predmetov ter njihove prostornine in cene, izpiše pa največjo možno skupno ceno predmetov v nahrbtniku.

Vhod

Prva vrstica vhoda vsebuje prostornino nahrbtnika ($V \in [0, 1000]$), druga število predmetov ($n \in [1, 1000]$), tretja prostornine predmetov ($v_i \in [1, 1000]$ za $i \in \{1, \dots, n\}$), četrta pa cene predmetov ($c_i \in [1, 1000]$ za $i \in \{1, \dots, n\}$).

Izhod

Izpišite največjo skupno ceno.

Testni primer 1

Vhod:

```
12
5
3 4 7 2 3
5 6 8 9 1
```

Izhod:

```
22
```

Izberemo prvi, tretji in četrti predmet.

Problem nahrbtnika 2

Naloga

Rešujemo enak problem kot pri nalogi Problem nahrbtnika 1, le da lahko nahrbtnik vsebuje kvečjemu en predmet z liho prostornino.

Vhod in izhod imata enako obliko kot pri nalogi Problem nahrbtnika 1.

Testni primer 1

Vhod:

```
12
5
3 4 7 2 3
5 6 8 9 1
```

Izhod:

```
20
```

Izberemo prvi, drugi in četrti predmet.

Vaje pri predmetu Programiranje 2

Teden 8: Strukture

Naloga

Podani sta sledeči deklaraciji (gl. datoteko `naloga.h`):

```
typedef struct _P0 {    // par predmet-ocena
    char predmet[4];    // enolična oznaka predmeta
    int ocena;          // ocena pri tem predmetu
} P0;

typedef struct _Student {    // podatki o študentu
    int vpisna;             // vpisna številka
    P0* po;                 // kazalec na začetek tabele parov predmet-ocena
    int stP0;               // število parov predmet-ocena
} Student;
```

V datoteki `naloga.c` napišite sledeče funkcije:

- `int poisciStudenta(Student** studentje, int stStudentov, int vpisna)`
V tabeli s `stStudentov` kazalci, na začetek katere kaže kazalec `studentje`, poišče kazalec na strukturo, ki predstavlja študenta z vpisno številko `vpisna`. Če takšen kazalec obstaja, naj funkcija vrne njegov indeks v tabeli, sicer pa naj vrne `-1`.
- `int poisciP0(Student* student, char* predmet)`
V tabeli parov predmet-ocena za podanega študenta (kazalec `student` kaže na strukturo, ki ga predstavlja) poišče par, v katerem je oznaka predmeta enaka `predmet`. Če takšen par obstaja, naj funkcija vrne njegov indeks v tabeli, sicer pa naj vrne `-1`.
- `int dodaj(Student** studentje, int stStudentov, int vpisna, char* predmet, int ocena)`

Tabelo s `stStudentov` kazalci, na začetek katere kaže kazalec `studentje`, obogati s podatkom, da je študent z vpisno številko `vpisna` na izpitu pri predmetu z oznako `predmet` prejel oceno `ocena`.

Če študent že ima svojo strukturo, naj jo funkcija posodobi (gl. naslednji odstavek), sicer pa naj ustvari novo strukturo in na konec tabele doda kazalec nanjo (lahko predpostavite, da je tabela dovolj velika). Kazalec `po` v novoustvarjeni strukturi naj kaže na začetek tabele z desetimi alociranimi pari predmet-ocena. Prvi od njih naj vsebuje podano oznako predmeta in podano oceno, ostalih pa ne inicializirajte.

Če v tabeli parov predmet-ocena za obstoječega študenta že obstaja par s podano oznako predmeta, naj funkcija v tem paru samo nastavi oceno na podano oceno, sicer pa naj na konec tabele doda nov par s podano oznako predmeta in podano oceno. Lahko predpostavite, da je tabela dovolj velika.

Funkcija naj vrne novo število elementov v tabeli, na začetek katere kaže kazalec `studentje`.

Vaje pri predmetu Programiranje 2

Teden 9: Povezani seznam

V datoteki `naloga.h` sta podani sledeči deklaraciji:

```
typedef struct Vozlisce Vozlisce;

struct Vozlisce {
    int podatek;           // podatek, ki ga hrani vozlišče
    Vozlisce* naslednje;   // kazalec na naslednje vozlišče (NULL, če ga ni)
};
```

Datoteko `naloga.c` dopolnite s funkcijami, navedenimi v nadaljevanju. Funkciji `vsotaI` in `vstaviUrejenoI` implementirajte iterativno, funkciji `vsotaR` in `vstaviUrejenoR` pa rekurzivno. Pri vseh funkcijah lahko ima kazalec `zacetek` tudi vrednost `NULL`; ta predstavlja prazen seznam.

- `int vsotaI(Vozlisce* zacetek);`
`int vsotaR(Vozlisce* zacetek);`

Vrne vsoto elementov povezanega seznama, na čigar začetno vozlišče kaže kazalec `zacetek`.

- `Vozlisce* vstaviUrejenoI(Vozlisce* zacetek, int element);`
`Vozlisce* vstaviUrejenoR(Vozlisce* zacetek, int element);`

Podani element vstavi na ustrezno mesto v naraščajoče urejenem povezanem seznamu, na čigar začetno vozlišče kaže kazalec `zacetek`, in vrne kazalec na (morebitno novo) začetno vozlišče obogatene seznama.

Vaje pri predmetu Programiranje 2

Teden 10: Besedilne datoteke

Najdaljša vrstica

Naloga

Napišite program, ki najdaljšo vrstico podane vhodne datoteke (skupaj z zaključnim znakom `\n`) prepiše v podano izhodno datoteko. Če je takih vrstic več, naj prepiše prvo med njimi. Nobena vrstica ni daljša od 10^6 znakov (vštevši znak `\n`).

Vhodna in izhodna datoteka sta podani kot argumenta ukazne vrstice. Prvi argument je pot do vhodne, drugi pa pot do izhodne datoteke.

Na primer, če program pokličemo kot

```
./najdaljsaVrstica vrba.txt naj.txt
```

pri čemer ima datoteka `vrba.txt` vsebino

```
O Vrba!  
srecna draga  
vas domaca
```

potem mora program ustvariti datoteko `naj.txt` z vsebino

```
srecna draga
```

Zlivanje

Naloga

Podanih je n vhodnih datotek, od katerih vsaka vsebuje naraščajoče urejeno zaporedje celih števil z intervala $[1, 10^9]$. Vsako število je zapisano v svoji vrstici.

Napišite program, ki ustvari izhodno datoteko s podanim imenom in vanjo v naraščajočem vrstnem redu prepiše vsa števila iz vseh vhodnih datotek. Vsako število naj bo zapisano v svoji vrstici.

Vhod

Prva vrstica (standardnega) vhoda vsebuje število $n \in [1, 10]$. Naslednjih n vrstic vsebuje imena posameznih vhodnih datotek, zadnja vrstica pa ime izhodne datoteke. Vsako ime je niz največ 100 znakov.

Izhod

Na (standardni) izhod naj program ne izpiše ničesar.

Testni primer 1

Vhod:

```
3
vh_01a.txt
vh_01b.txt
vh_01c.txt
rez_01.txt
```

Izhod je prazen, datoteke pa imajo sledečo vsebino:

vh_01a.txt:

```
4
5
7
10
15
```

vh_01b.txt:

```
3
6
9
18
```

vh_01c.txt:

```
1
10
12
20
```

rez01.txt:

```
1
3
4
5
6
7
9
10
10
12
15
18
20
```

Vaje pri predmetu Programiranje 2

Teden 11: Dvojiške datoteke

Barva pike

Naloga

Napišite program

`barva datoteka vrstica stolpec`

ki na standardni izhod izpiše vrstico oblike

`R G B`

pri čemer je R vrednost rdeče, G vrednost zelene, B pa vrednost modre komponente za slikovno piko, ki se na sliki, shranjeni v podani datoteki, nahaja v vrstici z indeksom *vrstica* in stolpcu z indeksom *stolpec*. Datoteka je zapisana v dvojiški različici formata PPM, kar pomeni, da ima sledečo zgradbo:

```
P6
širina višina
255
vsebina
```

Širina in višina slike sta zapisani kot števili v besedilni (ASCII) obliki. V bloku *vsebina* so v dvojiški obliki po vrsti zapisane vrednosti barvnih komponent posameznih pik — najprej rdeča, zelena in modra (v tem vrstnem redu) za prvo piko v prvi vrstici, nato rdeča, zelena in modra za drugo piko v prvi vrstici itd. Vsaka vrednost posamezne barvne komponente zasede po 1 bajt prostora.

Širina (w) in višina (h) vhodne slike pripadata intervalu $[1, 1000]$, parametra *vrstica* in *stolpec* pa intervalu $[0, h - 1]$ oziroma $[0, w - 1]$.

Primer

Sledi primer zagona programa ter vhodne slike in pripadajočega izhoda:

```
./barva slika01.ppm 1 0
```

slika01.ppm ($w = 3$, $h = 2$):



Izhod:

255 255 0

Pretvorba v sivine

Naloga

Napišite program

`pretvori vhod izhod`

ki barvno sliko, zapisano v datoteki *vhod* v formatu PPM (gl. prejšnjo nalogo), pretvori v sivinsko in jo shrani v datoteko *izhod* v formatu PGM. Datoteka v formatu PGM ima sledečo zgradbo:

```
P5
širina višina
255
vsebina
```

Širina in višina slike sta zapisani kot števili v besedilni (ASCII) obliki. V bloku *vsebina* so v dvojiški obliki po vrsti zapisane sivinske vrednosti posameznih pik — najprej za prvo piko v prvi vrstici, nato za drugo piko v prvi vrstici itd. Vsaka sivinska vrednost zasede po 1 bajt prostora.

Sivinsko vrednost pike izračunajte kot $\lfloor (30R + 59G + 11B) / 100 \rfloor$, kjer so R , G in B vrednosti posameznih barvnih komponent pike.

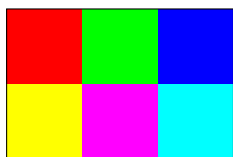
Širina in višina vhodne slike pripadata intervalu $[1, 1000]$.

Primer

Sledi primer zagona programa ter vhodne in pripadajoče izhodne slike:

```
./pretvori slika01.ppm rezultat01.pgm
```

slika01.ppm:



rezultat01.pgm:



Vaje pri predmetu Programiranje 2

Teden 12

Kalkulator

Naloga

Napišite program, ki prebere aritmetični izraz in izpiše njegovo vrednost.

Vhod

Na vhodu je zapisan niz največ 1000 znakov iz množice $\{0, 1, \dots, 9, +, -, *, /, (,)\}$. Niz predstavlja veljaven aritmetični izraz. Znaki $+$, $-$, $*$ in $/$ po vrsti predstavljajo seštevanje, odštevanje, množenje in celoštevilsko deljenje. Vsa števila v izrazu so enomestna. Znaka $+$ in $-$ nikoli ne nastopata v vlogi eniških operatorjev; izraz $5+(-3)$, denimo, ni mogoč.

Rezultati izraza in vseh njegovih podizrazov pripadajo intervalu $[-10^9, 10^9]$. Izraz ne vsebuje deljenja z ničlo.

Testni primeri 1–5 ne vsebujejo oklepajev.

Izhod

Izpišite vrednost izraza.

Testni primer 7

Vhod:

3+4*(5-2)

Izhod:

15

Napotek

Za potrebe naslednjega odstavka naj bo del izraza (operand ali operator) *svoboden*, če se ne nahaja znotraj oklepajev.

Najprej odstranite morebitne odvečne zunanje oklepaje. Zunanji par oklepajev je odveč, če noben del izraza ni svoboden. Nato poiščite najbolj desni svoboden operator $+$ ali $-$, če takega operatorja ni, pa najbolj desni svoboden operator $*$ ali $/$. Če niti tak operator ne obstaja, je izraz sestavljen iz enega samega števila, to število pa je seveda tudi njegov rezultat. V nasprotnem primeru pa rekurzivno izračunajte podizraz levo in podizraz desno od iskanega operatorja in rezultata s tem operatorjem združite.