

## zadaci

---

zadatak	kvadrat	svemir	jmbg
izvorni kôd	kvadrat.pas kvadrat.c kvadrat.cpp	svemir.pas svemir.c svemir.cpp	jmbg.pas jmbg.c jmbg.cpp
ulazni podaci	standardni ulaz	biblioteka	standardni ulaz
izlazni podaci	standardni izlaz	biblioteka	standardni izlaz
vremensko ograničenje (Athlon 64 3000+)	1 sekunda		
memorijsko ograničenje (heap)	32 MB		
memorijsko ograničenje (stack)	8 MB		
broj bodova	40	50	60
	150		

## kvadrat

---

Damirova najdraža igračka je magični kvadrat u kojem se nalazi **9 prirodnih brojeva** poredanih u 3 retka i 3 stupca.

Magični kvadrat ima svojstvo da su **zbrojevi brojeva** u svakom retku, svakom stupcu i na obje dijagonale **međusobno jednaki**, a svaki broj u njemu je manji od ili jednak 20000 (dvadeset tisuća).

Dok je Damir spavao, lopovi su ukrali **najviše tri broja** iz njegovog magičnog kvadrata.

Napišite program koji će za neki zadani (krnji) kvadrat odrediti brojeve koji nedostaju.

### ulazni podaci

U svakom od tri retka se nalaze tri cijela broja tj. opis magičnog kvadrata.

Na mjestima brojeva koji su ukradeni nalazi se **broj 0 (nula)**.

### izlazni podaci

Treba ispisati točno popunjeni magični kvadrat.

### test primjeri

#### ulaz

```
4 9 2
3 0 7
8 1 6
```

#### izlaz

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

#### ulaz

```
0 12 12
16 10 0
8 8 14
```

#### izlaz

```
6 12 12
16 10 4
8 8 14
```

#### ulaz

```
495 468 0
0 522 414
441 0 549
```

#### izlaz

```
495 468 603
630 522 414
441 576 549
```

Svemirski brod "Crveni Patuljak" je spletom čudnih okolnosti ostao bez svoje crne kutije. Ona se sada nalazi negdje u svemiru, a posada svemirskog broda pokušava doći do nje kako je netko drugi ne bi pronašao prije njih i čuo na njoj snimljene sočne razgovore kapetana i stjuardese.

Problem je u tome što signal koji crna kutija emitira nije dovoljno jak da bi posada odredila točnu poziciju kutije. Međutim, pri svakom pomaku broda precizni mjerni instrumenti mogu izmjeriti da li je signal jači ili slabiji nego prije te posada broda na taj način može znati **da li se brod približio kutiji ili se udaljio od nje**.

Svemir zamišljamo kao trodimenzionalni prostor koji se sastoji od  $N \times N \times N$  jediničnih polja. Svako polje je određeno s tri koordinate  $x$ ,  $y$  i  $z$ , a svaka od njih je prirodni broj manji od ili jednak  $N$ . Svemirski brod se na početku nalazi na koordinatama  $(1,1,1)$ , dok se crna kutija nalazi na nekoj drugoj poziciji.

Napišite program koji će pronaći crnu kutiju (tj. pomicanjem dovesti svemirski brod na poziciju gdje se ona nalazi) **s najviše 200 poziva** funkcije **Pomak**.

### biblioteka

Za rješavanje zadatka na raspolaganju je biblioteka funkcija **svemlib** koja sadrži dvije funkcije:

**DajN** - ova funkcija vraća vrijednost broja  $N$ ,  $2 \leq N \leq 1,000,000,000$  (milijardu).

```
function DajN : longint;  
int DajN(void);
```

**Pomak** - ova funkcija se poziva s tri argumenta - koordinate pozicije na koju želimo pomaknuti svemirski brod. Svi argumenti moraju biti prirodni brojevi manji od ili jednaki  $N$ .

```
procedure Pomak(x,y,z : longint) : longint;  
int Pomak(int x,int y,int z);
```

Funkcija **Pomak** pomiče brod na zadane koordinate i vraća jedan od sljedećih rezultata:

- 1 brod se približio kutiji
- 0 brod je ostao na istoj udaljenosti od kutije
- 1 brod se udaljio od kutije

Ukoliko pozivom ove funkcije svemirski brod dođe na polje na kojem se nalazi crna kutija, biblioteka će **regularno završiti izvođenje** vašeg programa.

### **Upute za natjecatelje u Pascal-u**

Na početku vašeg programa mora se nalaziti naredba `'uses svemlib'`.

Također, potrebno je preuzeti biblioteku s evaluacijskog sustava.

### **Upute za natjecatelje u C/C++**

Potrebno je preuzeti datoteke `svemlib.h` i `svemlib.o` s evaluacijskog sustava.

U RHIDE-u otvorite projekt (`Project->Open`) i nazovite ga `svemir`. U njega naredbom `Project->AddItem` dodajte datoteke `svemir.c` ili `svemir.cpp` (vaš program) i `svemlib.o`.

Izvršnu datoteku možete kreirati i kompajliranjem pomoću komandne linije (npr. za C)

```
gcc -o svemir.exe svemir.c svemlib.o.
```

Na početku vašeg programa mora se nalaziti naredba `'#include "svemlib.h"'`.

**Važno:** nemojte koristiti naredbu `Compile->BuildAll` (jer ona briše datoteku `svemlib.o`).

### **Testiranje**

Vaš program možete testirati tako da na standardni ulaz unesete broj `N` i koordinate na kojima se nalazi crna kutija, a biblioteka će na standardni izlaz ispisivati poruke o rezultatima pomicanja svemirskog broda.

### **test primjeri**

**ulaz**

```
2
2 2 2
```

**ulaz**

```
5
2 3 4
```

**ulaz**

```
10
8 1 5
```

## jmbg

---

Svaki stanovnik nepoznate prenapučene zemlje posjeduje svoj jedinstveni matični broj građana (JMBG). U toj zemlji JMBG je devetnaesteroznamenasti broj oblika:

DDMMGGGGAAAAAAAAAAK

gdje znamenke DD označavaju dan, MM mjesec, a GGGG godinu rođenja.

Godina rođenja je prirodni broj između 0001 i 9999. Godina je prijestupna ako je djeljiva s 4, a nije djeljiva sa 100 ili ako je djeljiva s 400.

Za znamenke označene s A smatramo da su proizvoljne, a K je kontrolna znamenka dobivena po sljedećem algoritmu:

1. Označimo sve znamenke u JMBG-u osim zadnje redom sa Z1 do Z18
2.  $S = (10*Z1 + 9*Z2 + 8*Z3 + \dots + 2*Z9 + 10*Z10 + 9*Z11 + 8*Z12 + \dots + 2*Z18) \bmod 19$
3. ako je  $S \leq 9$  tada je  $K = S$ , inače je  $K = 19 - S$

U lovu na kriminalce policija često među sobom razmjenjuje njihove JMBG-ove. No, kako su njihovi voki-tokiji prilično zastarjeli, često se desi da policajac koji prima informaciju i zapisuje JMBG na papir ne čuje neke znamenke pa ih na papiru označi s 'X'. Tako zapisan JMBG policajac predaje svom kolegi informatičaru.

Napišite program koji će odrediti koliko je **različitih važećih JMBG-ova** moguće dobiti iz zapisa kojeg je dobio informatičar.

### ulazni podaci

U prvom i jedinom retku se nalazi niz od 19 znakova sastavljen od znamenaka '0'..'9' i znaka 'X'.

### izlazni podaci

U prvi i jedini redak treba ispisati traženi broj različitih važećih JMBG-ova iz teksta zadatka.

**Napomena:** rješenje će stati u 64-bitni cjelobrojni tip podataka (int64 u Pascalu, long long u C/C++).

### test primjeri

ulaz	ulaz	ulaz
XX0220051234567890X	XXXX200577XXXXXXXX7X	0XX52X0512X456X8903
izlaz	izlaz	izlaz
28	3650000000	946