

## Zadaci

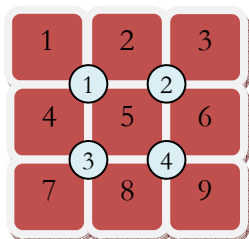
---

zadatak	rotacija	stat3000	dostava
izvorni kôd	rotacija.pas rotacija.c rotacija.cpp	stat3000.pas stat3000.c stat3000.cpp	dostava.pas dostava.c dostava.cpp
izvršna datoteka	rotacija.exe	stat3000.exe	dostava.exe
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		
broj bodova	20	30	50
	100		

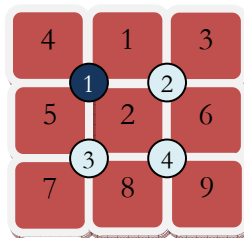
# ROTACIJA

Ivica i Marica su prošle godine bili jako dobri pa im je zla vještica za Novu godinu darovala jednu jako zanimljivu igračku. Igračka je ploča koja se sastoji od **devet polja** označenih brojevima od 1 do 9, poredanih u tri reda i tri stupca.

Ploča također sadrži **četiri tipke** označene brojevima od 1 do 4 kao na donjim slikama. Pritiskom na pojedinu tipku, četiri polja koje pritisnuta tipka dodiruje zakreću se u smjeru kazaljke na satu. Pritisak bilo koje tipke nazivamo **potezom**.



Početno stanje igračke.



Stanje nakon pritiska tipke 1.

Igra koju Ivica i Marica igraju započinje tako da Marica stavi povez preko očiju, a Ivica postavi igračku u početno stanje prikazano na slici. Zatim Ivica nasumce pritišće tipke pritom izgovarajući brojeve na pritisnutim tipkama. Maričin je zadatak odrediti konačno stanje igračke nakon svih Ivičinih poteza.

Vaš je zadatak da napišete program koji će to učiniti umjesto nje.

## Ulazni podaci

U prvom redu ulaza nalazi se prirodni broj  $N$  ( $1 \leq N \leq 20$ ), broj Ivičinih poteza.

U sljedećem redu nalazi se  $N$  brojeva odvojenih razmacima, oznake pritisnutih tipaka, redom kojim ih je Ivica pritisnulo.

## Izlazni podaci

U tri reda treba ispisati po tri broja odvojena razmacima, stanje igračke nakon svih poteza.

## Primjeri test podataka

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
1	3	4
1	1 3 2	1 1 1 1
<b>izlaz</b>	<b>izlaz</b>	<b>ulaz</b>
4 1 3	4 5 1	1 2 3
5 2 6	7 6 3	4 5 6
7 8 9	8 2 9	7 8 9

# STAT3000

---

STAT3000 je najnoviji sustav za praćenje grešaka u proizvodnji plišanih igračka u tvornici.

Sustav prima dvije vrste naredbi, **prijave grešaka** i **statističke upite**. Svaka naredba sadrži i **vremensku oznaku**, vrijeme u danu kad je naredba zadana. Vremenska oznaka je oblika "hh:mm:ss", gdje prvi dio predstavlja sate (00 do 23), drugi dio minute (00 do 59), a treći sekunde (00 do 59).

Svaki put kad se postavi statistički upit, sustav ispisuje tri broja, i to redom:

- Ukupan broj grešaka prijavljenih od početka rada sustava;
- Broj grešaka prijavljenih u zadnjih sat vremena (uključujući greške prijavljene prije točno sat vremena);
- Broj grešaka prijavljenih u zadnjoj minuti (uključujući greške prijavljene prije točno minutu).

Vremenske oznake ne predstavljaju nužno vremena u istom kalendarskom danu. Ako je vremenska oznaka neke naredbe manja od vremenske oznake naredbe koja joj neposredno prethodi, onda se smatra da je novo vrijeme dio sljedećeg dana. Tako je npr. između dvaju susjednih naredbi s vremenskim oznakama 23:00:00 i 00:30:00 prošao točno 1 sat i 30 minuta.

Zadan je niz naredbi koje sustav treba obraditi. Napišite program koji simulira rad sustava te ispisuje odgovor na svaki statistički upit.

## Ulazni podaci

U prvom redu ulaza nalazi se prirodni broj  $N$  ( $1 \leq N \leq 50$ ), broj naredbi.

Svaki od sljedećih  $N$  redova sadrži jednu naredbu oblika "P hh:mm:ss" (prijava greške) ili "U hh:mm:ss" (statistički upit).

Nijedan par susjednih naredbi neće imati istu vremensku oznaku.

## Izlazni podaci

Za svaki statistički upit potrebno je u jedan red ispisati tri broja odvojena razmacima, kako je opisano u tekstu zadatka.

## Primjeri test podataka

**ulaz**

```
4
P 09:20:00
P 09:30:00
U 10:00:00
U 10:25:00
```

**izlaz**

```
2 2 0
2 1 0
```

**ulaz**

```
5
P 23:59:40
U 00:00:40
P 00:59:40
U 00:59:41
U 01:00:41
```

**izlaz**

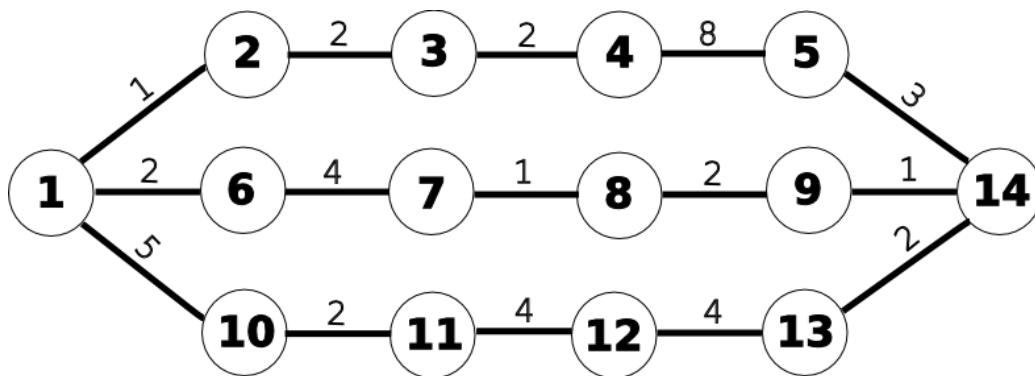
```
1 1 1
2 1 1
2 1 0
```

## DOSTAVA

Mirko je nedavno, uz potpunu podršku i odobravanje svojih roditelja, odlučio prekinuti svoje daljnje školovanje i potražiti posao u lokalnom trgovačkom poduzeću "Uzetćutisve".

Njegov posao je vožnja kamiona punog kataloga i dostavljanje istih zainteresiranim mušterijama. On je za trajanja radnog dana u stalnom kontaktu sa svojim šefom, koji mu **nakon** svake obavljene dostave dojavljuje lokaciju sljedeće. Mirko ne zna unaprijed kamo će ga šef poslati, ali može što brže obaviti zadane zadatke pametnim odabirom puta. Zato je nabavio plan područja grada u kojem radi.

Mirko je zadužen za dostavu u **tri usporedne ulice** koje na svojim krajevima završavaju s **dva trga** u kojima se sve tri sijeku. U svakoj se ulici nalazi **jednaki broj** zgrada u kojima prebivaju potencijalne mušterije. Također, trгови su uvijek prepuni potencijalnih mušterija pa šef ponekad Mirka upućuje i na trgove.



Slika ilustrira drugi primjer.

Poznata su vremena potrebna kamionu da prijeđe put između svake dvije susjedne zgrade (unutar iste ulice), te od oba trga do prve i posljednje zgrade u svakoj ulici.

Napišite program koji, za zadane lokacije koje Mirko mora obići u jednom radnom danu, pronalazi koliko mu je najmanje vremena potrebno za to.

### Ulazni podaci

U prvom redu ulaza nalazi se prirodni broj  $N$  ( $1 \leq N \leq 100\,000$ ), broj zgrada u svakoj od tri ulice.

U svakom od sljedeća tri retka nalazi se po  $N+1$  prirodnih brojeva, redom vremena potrebna za prolazak od trga do prve zgrade u ulici, od prve do druge zgrade, itd., sve do vremena potrebnog za dolazak od posljednje zgrade u ulici do drugog trga. Svaki od tih brojeva bit će manji ili jednak od 10000.

U sljedećem retku nalazi se broj  $K$  ( $1 \leq K \leq 100\,000$ ), broj lokacija na koje Mirko treba dostaviti katalog. Nakon toga slijedi  $K$  brojeva, oznake tih lokacija, redom kojim ih treba obići. Moguće lokacije su sve zgrade ili jedan od dva trga.

Trгови su numerirani brojevima 1 i  $3 \cdot N + 2$ . Zgrade u prvoj ulici označene su brojevima od 2 do  $N+1$ , u drugoj od  $N+2$  do  $2 \cdot N + 1$ , te u trećoj od  $2 \cdot N + 2$  do  $3 \cdot N + 1$ , kao na slici.

Mirko se na početku nalazi na trgu označenom brojem 1.

# DOSTAVA

---

## Izlazni podaci

U jedini redak izlaza ispišite najmanje ukupno vrijeme potrebno za dostavljanje svih kataloga.

**Napomena:** Koristite 64-bitni cjelobrojni tip podataka (long long u C-u, odnosno Int64 u Pascalu).

## Bodovanje

U test podacima vrijednim 70% bodova ulazni podaci N i K bit će najviše 100.

## Primjeri test podataka

**ulaz**

```
2
1 2 1
3 3 3
1 2 1
5
7 4 3 2 5
```

**izlaz**

23

**ulaz**

```
4
1 2 2 8 3
2 4 1 2 1
5 2 4 4 2
5
8 11 4 14 1
```

**izlaz**

53