

# Županijsko natjecanje iz informatike

Srednja škola  
Prva podskupina (1. i 2. razred)

10. veljače 2017.

## Zadaci

Ime zadatka	Vremensko ograničenje	Memorijsko ograničenje	Broj bodova
<b>Nijansa</b>	1 sekunda	512 MiB	30
<b>Gitara</b>	1 sekunda	512 MiB	45
<b>Cjevovod</b>	1 sekunda	512 MiB	55
<b>Voda</b>	10 sekundi	512 MiB	70
<b>Ukupno</b>			200

## Zadatak: Nijansa

U računalnoj grafici postoji više različitih načina zapisivanja boja. U takozvanom *RGB prostoru* je svaka boja definirana pomoću tri cijela broja  $R$ ,  $G$  i  $B$  između 0 i 255 koji opisuju udio osnovnih boja — crvene, zelene i plave (eng. *red*, *green*, *blue*). Jedna od alternativa je takozvani *HSV prostor* u kojem je boja definirana pomoću tri broja  $H$ ,  $S$  i  $V$  koji opisuju *nijansu*, *zasićenost* i *vrijednost* boje (eng. *hue*, *saturation*, *value*). Ovdje opisujemo pojednostavljeni postupak kojim se računa nijansa  $H$  na temelju udjela  $R$ ,  $G$  i  $B$ .

Pretpostavimo da su brojevi  $R$ ,  $G$  i  $B$  međusobno različiti, nijansa  $H$  je realni broj između 0 i 360 koji računamo na sljedeći način:

1.  $C_{max}$  je najveća od vrijednosti  $R$ ,  $G$  i  $B$ .
2.  $C_{min}$  je najmanja od vrijednosti  $R$ ,  $G$  i  $B$ .
3.  $\Delta$  je njihova razlika  $C_{max} - C_{min}$ .
4.  $H$  računamo prema formuli koja ovisi od tome koji je od brojeva  $R$ ,  $G$  i  $B$  najveći:
  - (a)  $H = 60 \left( \frac{G-B}{\Delta} \right)$  kada je  $C_{max} = R$ .
  - (b)  $H = 60 \left( 2 + \frac{B-R}{\Delta} \right)$  kada je  $C_{max} = G$ .
  - (c)  $H = 60 \left( 4 + \frac{R-G}{\Delta} \right)$  kada je  $C_{max} = B$ .
5. Ako je  $H$  ispao negativan, onda ga uvećamo za 360.

Za zadanu boju opisanu pomoću udjela  $R$ ,  $G$  i  $B$  izračunajte njenu nijansu  $H$ .

### Ulazni podaci

Prvi red sadrži tri međusobno različita cijela broja  $R$ ,  $G$  i  $B$  ( $0 \leq R, G, B \leq 255$ ) — udjele osnovnih boja.

### Izlazni podaci

U prvi red ispišite traženu vrijednost  $H$ .

### Bodovanje

- Rješenje za pojedini test podatak se smatra ispravnim ukoliko odstupa od službenog rješenja za najviše 0.1.
- U test podacima vrijednim 50% bodova vrijedi  $R > G > B$ .

### Primjeri test podataka

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
200 100 50	223 20 220	0 215 255	50 60 40
<b>izlaz</b>	<b>izlaz</b>	<b>izlaz</b>	<b>izlaz</b>
20.0	300.886699507	189.411764706	90.0

## Zadatak: Gitara

*Tablatura* je način zapisivanja glazbe koji opisuje redosljed i način trzanja žica na gitari. Žicu na gitari je moguće trznuti ili uz stiskanje jednog od 19 polja na vratu gitare (označenih brojevima od 1 do 19) ili bez stiskanja polja. Dakle, određenu žicu možemo trznuti na 20 načina i tako proizvesti 20 različitih tonova.

Tonove označavamo na sljedeći način: *kromatska ljestvica* se sastoji od *oktava* redom označenih prirodnim brojevima od jedan na dalje. Jedna oktava se sastoji od točno dvanaest tonova označenih redom sa nizovima znakova: C, C#, D, D#, E, F, F#, G, G#, A, A#, H, a nakon svakog od tih nizova obavezno pišemo redni broj oktave. Dakle "C1", "D#1" i "H5" su oznake tonova. Svaki ton u oktavi je za jedan *polustupanj* viši od prethodnog, a prvi ton oktave je za jedan polustupanj viši od zadnjeg tona prethodne oktave. Pa je tako, primjerice, "D#1" za tri polustupnja viši od "C1" dok je "C6" za jedan polustupanj viši od "H5".

*Osnovni ton* žice je ton koji dobijemo kada žicu trznemo bez stiskanja polja, a gitara je napravljena tako da ako žicu trznemo uz stiskanje polja  $k$  onda dobijemo ton koji je za točno  $k$  polustupnja viši od osnovnog tona žice. Naša gitara se sastoji od 6 žica označenih brojevima od 1 do 6, čiji su osnovni tonovi redom "E4", "H3", "G3", "D3", "A2", "E2". Na primjer, trazanjem žice 5 uz stiskanje polja 4 dobijemo ton "C#3".

Pretpostavimo da opisujemo glazbeno djelo duljine  $n$  taktova (takt je jedinična mjera vremena) označenih brojevima od 1 do  $n$ . Tablatura tog djela je tablica znakova koje se sastoji od 6 redaka i  $n$  stupaca. Ukoliko je u  $i$ -tom taktu potrebno trznuti  $j$ -tu žicu onda u  $i$ -ti znak u  $j$ -tom retku tablature zapisujemo:

- Znamenku "0" ukoliko se ne stiska polje prilikom trzanja.
- Broj  $k$  ukoliko se stiska polje  $k$  prilikom trzanja.
  - Ukoliko je  $k$  dvoznamenkasti broj, prva znamenka se upisuje u  $i$ -ti znak, a druga u sljedeći znak u  $j$ -tom retku (dakle, druga znamenka se upisuje u stupac koji zapravo pripada taktu  $i + 1$ ).

Ukoliko u određeno mjesto u tablici nismo upisali znamenku onda tamo zapisujemo znak "-" (minus). Prilikom zapisa tablature, prije prvog i nakon zadnjeg stupca dodajemo stupac sastavljen od znakova "|".

Za zadanu tablaturu, odredite niz tonova koji se odsvira prilikom izvođenja glazbenoga djela redom od početka do kraja. Naše djelo je jednostavno pa možete pretpostaviti da se u svakom taktu odsvira najviše jedan ton. Dodatno možete pretpostaviti da na svakoj pojedinoj žici, nakon što se odsvira ton u nekom taktu, dolaze još barem dva takta u kojima se ništa ne svira na toj žici. Stoga se oznake polja u tablaturi koje pripadaju različitim trzajima iste žice ne preklapaju niti dotiču.

### Ulazni podaci

Ulaz se sastoji od šest redova jednake duljine. U  $j$ -tom redu se nalazi  $j$ -ti redak tablature zajedno s graničnim znakovima. Prvi i zadnji znak svakog retka je "|" (vertikalna crta, ASCII 124), dok je svaki ostali znak ili znamenka ili znak "-" (minus). Djelo se sastoji od najviše 100 taktova ( $n \leq 100$ ), što znači da je svaki red ulaza duljine najviše 102 znaka. Možete pretpostaviti da djelo sadrži najmanje jedan ton.

### Izlazni podaci

Ispišite  $m$  redova gdje je  $m$  ukupan broj tonova u djelu. U  $k$ -ti red ispišite oznaku  $k$ -tog po redu tona koji se odsvira prilikom izvođenja djela. Oznaka tona se sastoji od velikog slova, nakon kojeg kod nekih tonova dolazi znak "#" (ASCII 35), te prirodnog broja. Između elemenata oznake jednog tona ne dolaze razmaci.

### Bodovanje

- U test podacima vrijednim 30% bodova u ulazu se ne pojavljuje niti jedna znamenka osim "0".
- U dodatnim test podacima vrijednim 40% bodova pojavljuju se samo jednoznamenaste oznake polja.

### Primjeri test podataka

**ulaz**

```
|---|  
|0--|  
|---|  
|---|  
|---|  
|---|  
|---|
```

**izlaz**

H3

**ulaz**

```
|---0-----|  
|--0-0-----|  
|-0---0-----|  
|-----0----|  
|-----0---|  
|0-----0--|
```

**izlaz**

E2  
G3  
H3  
E4  
H3  
G3  
D3  
A2  
E2

**ulaz**

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

**izlaz**

C5  
G#4  
D#4  
A#4  
E4  
H3  
A#3  
E3  
A#2  
F2  
F#3  
C#5

**ulaz**

```
|-----5-----10-|  
|-----15-----11--|  
|---16-----10----|  
|--17--19-17-----|  
|-18-----11-----|  
|19-----|
```

**izlaz**

H3  
D#4  
G4  
H4  
D5  
A4  
A4  
G4  
G#3  
F4  
A#4  
D5

## Zadatak: Cjevovod

*Assembler* je programski jezik niže razine vrlo blizak strojnom jeziku računala. Naš procesor sadrži 10 registara označenih redom s  $r_0, r_1, \dots, r_9$ , a razmatramo jednostavan podskup jezika koji se sastoji od instrukcija oblika “*op ra rb rc*”. Ova instrukcija primjenjuje operaciju *op* na vrijednosti registara *rb* i *rc* te rezultat operacije sprema u registar *ra*. Registar *ra* je *cilj*, a registri *rb* i *rc* *izvori* u toj instrukciji.

Ukoliko instrukcija *B* dolazi u programu negdje nakon instrukcije *A*, a barem jedan od izvora u *B* je jednak cilju u *A* onda kažemo da *B* *ovisi* o *A*. Primjerice u programu “*add r3 r2 r4; add r4 r0 r9; add r5 r2 r3*” treća instrukcija koristi kao izvor registar *r3* koji je cilj operacije u prvoj instrukciji pa stoga treća instrukcija ovisi o prvoj instrukciji.

Procesor pomoću svog cjevovoda (eng. *pipeline*) može obrađivati više instrukcija u isto vrijeme što ubrzava izvršavanje programa, ali dovodi do problema s ovisnim instrukcijama. Za naš procesor vrijedi da ako instrukcija *B* ovisi o instrukciji *A* onda u programu treba postojati *barem 5 instrukcija* između instrukcija *A* i *B* kako bi program bio *ispravan*.

Ako program nije ispravan, možemo ga modificirati tako da na proizvoljna mjesta u programu dodamo proizvoljan broj takozvanih NOP instrukcija koje ništa ne rade. Nisu dozvoljene druge modifikacije programa poput brisanja ili mijenjanja redoslijeda instrukcija. Odredite duljinu najkraćeg ispravnog programa koji se na opisani način može dobiti modifikacijom zadanog programa.

### Ulazni podaci

U prvom redu nalazi se prirodni broj  $n$  ( $1 \leq n \leq 1000$ ) — broj instrukcija u programu. U  $j$ -tom od sljedećih  $n$  redova nalaze se redom oznaka registra cilja te oznake dvaju registara izvora u  $j$ -toj instrukciji. Oznaka registra je niz od točno dva znaka oblika “*rk*” (bez razmaka) gdje je  $k$  znamenka između 0 i 9. Oznake registara su odvojene točno jednim razmakom.

### Izlazni podaci

U prvi red ispišite traženu najmanju duljinu programa.

### Primjeri test podataka

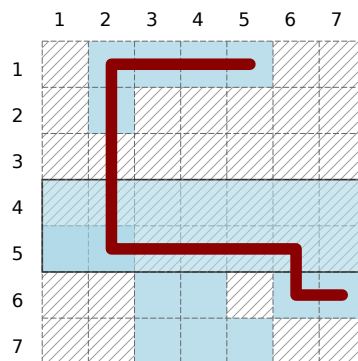
<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
3	7	7
r3 r2 r4	r0 r0 r0	r5 r4 r6
r4 r0 r9	r0 r1 r2	r3 r1 r2
r5 r3 r2	r1 r2 r3	r4 r5 r6
<b>izlaz</b>	r3 r2 r3	r9 r3 r2
7	r2 r2 r2	r4 r5 r6
	r0 r0 r0	r9 r9 r3
	r1 r9 r8	r1 r5 r6
	<b>izlaz</b>	<b>izlaz</b>
	9	15

**Pojašnjenje prvog primjera:** Treća instrukcija ovisi o samo prvoj, dok prva i druga instrukcija ne ovise o ničemu. Jedan najkraći ispravn program dobijemo umetanjem 4 NOP instrukcije prije treće instrukcije: “*add r3 r2 r4; add r4 r0 r9; NOP; NOP; NOP; NOP; add r5 r2 r3*”.

## Zadatak: Voda

Ledolomac se na djelomično smrznutom Dunavu mora probiti od točke  $A$  do točke  $B$ . Mapa rijeke je kvadratna mreža znakova koja se sastoji od  $n$  redaka označenih brojevima od 1 do  $n$  odozgo prema dolje, te  $n$  stupaca označenih brojevima od 1 do  $n$  slijeva nadesno. Neka polja u mapi su zaleđena (označena znamenkom "1") dok ostala nisu (označena znamenkom "0"). Ledolomac se može micati u četiri smjera (gore, dolje, lijevo ili desno) te se može kretati i preko slobodnih i preko zaleđenih polja. Međutim, kretanje po zaleđenim poljima je puno teže pa posada želi naći put od  $A$  do  $B$  koji sadrži što manje takvih polja.

Dodatno, posada ima na raspolaganju jednu kemijsku bombu koju mogu precizno lansirati bilo gdje na rijeku. Pomoću bombe *dosega*  $k$  je moguće otopiti sva zaleđena polja unutar jednog pravokutnog područja koje se sastoji od točno  $k$  uzastopnih redaka mape.



Slika 1: Rješenje jedinog scenarija iz prvog primjera test podataka

Zadana je mapa rijeke te  $q$  različitih scenarija. U svakom scenariju  $j$  su zadana polja  $A_j$  i  $B_j$  te doseg bombe  $k_j$ . Za svaki scenarij odredite najmanji mogući broj zaleđenih polja koje ledolomac treba prijeći na zadanoj mapi kako bi se probio od  $A_j$  do  $B_j$ , uz upotrebu jedne bombe dosega  $k_j$ .

### Ulazni podaci

U prvom redu nalazi se prirodni broj  $n$  ( $1 \leq n \leq 2000$ ) — dimenzije mape. U  $j$ -tom od sljedećih  $n$  redova nalazi se niz od točno  $n$  znakova "0" ili "1" —  $j$ -ti redak mape.

U sljedećem redu nalazi se prirodni broj  $q$  ( $1 \leq q \leq 5$ ) — broj scenarija. U  $j$ -tom od sljedećih  $q$  redova nalazi se pet prirodnih brojeva  $k, r_{A_j}, s_{A_j}, r_{B_j}$  i  $s_{B_j}$  ( $1 \leq k_j \leq n, 1 \leq r_{A_j}, s_{A_j}, r_{B_j}, s_{B_j} \leq n$ ) — doseg bombe te koordinate (broj retka pa broj stupca) polja  $A_j$  i polja  $B_j$  u  $j$ -tom scenariju. U svakom scenariju su polja  $A_j$  i  $B_j$  međusobno različita te niti jedno od njih nije zaleđeno.

### Izlazni podaci

Ispišite  $q$  redova. U  $j$ -ti red ispišite traženi minimalni broj zaleđenih polja u  $j$ -tom scenariju.

### Bodovanje

- U test podacima vrijednim 20% bodova vrijedi  $n \leq 20$
- U dodatnim test podacima vrijednim 20% bodova vrijedi  $n \leq 200$ .

### Primjeri test podataka

**ulaz**

7  
1000011  
1011111  
1111111  
1111111  
0011111  
1100100  
1100011  
1  
2 1 5 6 7

**izlaz**

1

**ulaz**

8  
01111100  
10011101  
11111110  
11011111  
11110101  
11010110  
11111101  
01110010  
4  
2 1 1 8 8  
2 1 8 8 1  
3 1 1 8 8  
4 1 8 8 1

**izlaz**

3  
2  
2  
1

**Pojašnjenje prvog primjera:** Ukoliko posada lansira bombu tako da se otopi sav led u četvrtom i petom retku onda postoji put od  $A$  do  $B$  koji prolazi kroz samo jedno zaleđeno polje.