



20. veljače 2013. od 14:30 do 16:30

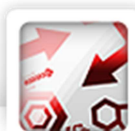
Infokup 2013

Županijsko natjecanje / Osnovna škola (7. razred)
Algoritmi (Basic/Pascal/C/C++)

Sadržaj

Zadaci.....	1
Zadatak: Alka	2
Zadatak: Gandalf	3
Zadatak: Snake	4





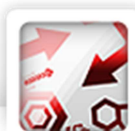
Zadaci

U tablici možete pogledati ograničenja za zadatke:

Zadatak	Alka	Gandalf	Snake
Vremensko ograničenje	5 sekundi	5 sekundi	5 sekundi
Broj bodova	40	70	90
Ukupno bodova		200	

NAPOMENE:

- kao rješenje zadatka treba predati njegov izvorni kod koji mora biti spremljen u obliku *ime_zadatka.nastavak* (.bas ili .sb ili .pas ili .c ili .cpp);
- vaše rješenje će se testirati na službenim test primjerima. U pravilu se prilikom evaluacije neće gledati vaš izvorni kod već samo njegova izvršna (.exe) verzija;
- ako se pri izvršavanju programa, na nekom test primjeru dogodi pogreška, tada taj primjer nosi 0 bodova;
- natjecatelji koji zadatke rješavaju u QuickBasicu i Small Basicu trebaju paziti na učitavanje ulaznih podataka. Ako je u zadatku predviđeno učitavanje više podataka u istom retku, tada se oni učitavaju jedan ispod drugog redoslijedom s lijeva na desno iz retka.

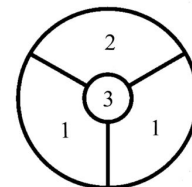


Zadatak: Alka

40 bodova

Sinjska alka je tradicionalno konjičko natjecanje u kojem jahači (alkari) na konju, u punom galopu, 3 metra dugačkim kopljem gađaju željeznu alku obješenu na konopu preko trkališta. Ustanovljena je 1715. godine u spomen na pobjedu Sinjana nad turskim osvajačima, a tradicionalno se održava prve nedjelje u kolovozu.

Cilj natjecanja je u **tri trke** (pokušaja) sakupiti **što više punata** (bodova) gađajući alku, tj. željezni kolut kao na slici. Pogodak u mali srednji kolut (sridu) donosi **3 punta**, u gornji pregradak **2**, a pogodak u jedan od dvaju pregradaka po **1 punat**. Kada alkar promaši alku, kaže se da je pogodio „**u ništa**“ i dobiva 0 punata. Pobjednik je onaj alkar koji sakupi najviše punata. Ako više alkara ima najviše, a jednako bodova, tada se pristupa tzv. **pripetavanju**, tj. trčanju dodatnih trka nakon kojih će se odrediti jedan pobjednik.



Među svim dodatnim pravilima posebno je zanimljivo ono po kome u trećoj trci mogu sudjelovati **samo oni koji još uvijek imaju priliku postati pobjednikom**.

Napiši program koji na osnovi broja bodova svih alkara nakon druge trke, određuje i ispisuje koliko je natjecatelja **sudjelovalo u trećoj trci**.

ULAZNI PODACI

U prvom retku nalazi se prirodan broj **N** ($1 \leq N \leq 10$), broj alkara u natjecanju.

U sljedećih **N** redaka nalazi se po jedan cijeli broj **P_i** ($0 \leq P_i \leq 6$), broj osvojenih punata svakog od **N** alkara.

IZLAZNI PODACI

U jednom retku treba ispisati broj alkara koji su sudjelovali u trećoj trci.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
5	4	5
5	6	3
4	1	3
6	0	3
3	0	3
2		3
izlaz	izlaz	izlaz
4	1	5

Pojašnjenje prvog test primjera: U trećoj trci nije nastupio alkar koji je nakon dvije trke sakupio samo 2 boda.



Zadatak: Gandalf

70 bodova

Oduvijek je mladim Hobitima strogo zabranjeno upotrebljavati razna pirotehnička sredstva. Zbog toga se uvijek razvesele kada im u posjet dođe Gandalf, čarobnjak iz Međuzemlja. Kako je čarobnjak, on smije koristiti pirotehnička sredstva i tako uveseljavati hobitsku djecu eksplozijama petardi i vatrometima u obliku zmaja.

Gandalf sam izrađuje svoje pirotehničke naprave. Svaka ima displej na kojem se odbrojava preostalo vrijeme u sekundama do eksplozije. Posebno je zanimljiv taj način odbrojanja.

Za početak, odbrojanje ne kreće uvijek od istog broja i nije kao u klasičnom slučaju da se svake sekunde preostalo vrijeme smanji za jednu sekundu. Odbrojanje se vrši tako da se svake sekunde broju koji predstavlja preostalo vrijeme, **od svake njegove pojedinačne znamenke, oduzme jedan**. Ukoliko se znamenka nula smanji za jedan, tada ta znamenka iščezne te više ne postoji. Od preostalih znamenki nastaje novi broj koji se zatim prikaže na displeju. Vodeće nule u broju se ne prikazuju. Odbrojanje završava kada se na displeju pojavi **broj** nula. Nakon toga slijedi detonacija.

Npr., ako odbrojanje počinje od vrijednosti 345, tada imamo sljedeću sekvencu pri odbrojanju: 345, 234, 123, 12, 1, 0. Ili, recimo za broj 324 imali bismo 324, 213, 102, 1, 0.

Napiši program koji za početnu vrijednost od koje počinje odbrojanje ispisuje slijed brojeva pri odbrojanju tj. **cijelu sekvencu odbrojanja** od polaznog broja do završne vrijednosti nula.

ULAZNI PODACI

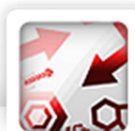
U prvom retku nalazi se jedan prirodan broj S ($1 \leq S \leq 99999$), početni broj od kojeg kreće odbrojanje.

IZLAZNI PODACI

Niz brojeva, svaki u svom redu, cijela sekvencu odbrojanja od polaznog broja do nule.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
342	123	604
izlaz	izlaz	izlaz
342	123	604
231	12	53
120	1	42
1	0	31
0		20
		1
		0



Zadatak: Snake

90 bodova

Lucija za zadaću iz informatike treba napraviti pojednostavljenu verziju igrice Snake. Profesor će zadati **početno stanje igrice** i **poteze** koje zmija treba napraviti, a njezin program mora prikazati konačno stanje igrice.

Igrica je zadana tablicom stanja koja ima **N** redaka i **M** stupaca te je popunjena s 4 vrste znakova: '.' - slobodno mjesto, 'X' - mjesto koje zauzima zmijino tijelo, '#' - mjesto na kojem se nalazi prepreka i '*' - mjesto na kojem se nalazi hrana.

Potezi su predstavljeni stringom duljine **P** koji se sastoji od 3 vrste znakova: 'L' - okretanje glave ulijevo, 'D' - okretanje glave udesno i 'R' - nema okretanja glave.

U **početnom stanju** zmija zauzima jedno polje i glava joj je **okrenuta prema** desnom kraju tablice. Zmija se zatim pomiče u skladu sa zadanim potezima. Kada je na nekom potezu, zmija ovisno o vrsti znaka, okreće ili ostavlja glavu u istom smjeru, a zatim se pokušava pomaknuti na **sljedeće polje** - polje prema kojem je okrenuta glava. Ako se uspješno pomakne prelazi na idući potez, inače igra završava.

Kada zmija dođe na polje sa hranom, pojede ju i pritom poraste. Tijelo joj se produži tako da **rep ostaje gdje je bio**, a glava se pomakne na polje na kojem je bila hrana. Ta hrana sada više ne postoji. Kada zmijino tijelo (rep) napusti neko mjesto, ono postaje **slobodno**.

Igra završava kada se odrade svi potezi ili u slučaju da se zmija ne može pomaknuti na sljedeće polje. To se događa: kada se na sljedećem polju nalazi **prepreka**, kada je sljedeće polje **izvan tablice** (oko tablice nalazi se zid) ili kada se na sljedećem polju nalazi **zmijino tijelo**.

Kada igra završi, Lucija **preostale poteze** (ako ih ima) mora **zanemariti** i ispisati konačno stanje. Budući da nije sigurna kako bi to sama napravila, traži pomoć!

ULAZNI PODACI

U prvom retku nalaze se dva prirodna broja: **N** ($1 \leq N \leq 20$), broj redaka i **M** ($1 \leq M \leq 20$), broj stupaca početne tablice stanja.

U sljedećih **N** redaka nalazi se po **M** znakova koji opisuju početni izgled tablice stanja.

U sljedećem retku nalazi se jedan prirodan broj **P** ($1 \leq P \leq 100$), broj zadanih poteza.

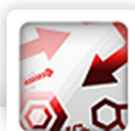
U posljednjem retku nalazi se string duljine **P** koji opisuje poteze koje zmija treba napraviti.

IZLAZNI PODACI

U svakom od **N** redova treba ispisati **M** znakova koji prikazuju konačni izgled tablice stanja.

BODOVANJE

U test primjerima vrijednim 36 bodova zmija nikad **neće pojesti** hranu - neće rasti. U test primjerima vrijednim 18 bodova zmija će **pojesti točno jednu** hranu.



PRIMJERI TEST PODATAKA

ulaz 4 5X*..#.. 6 DLRDLL	ulaz 6 5 X...# #*..# #...* #*#* #..# #*..# 9 RDRLRDLRD	ulaz 5 5 ..**. ...X. .**.. 8 LLLRDDDR
izlaz*.. ...X ..#..	izlaz# #..# #.XX. #*#XX #..# #*..#	izlaz ..X.. .XX.. .XX..

Pojašnjenje prvog test primjera: Prikazano je stanje igrice po potezima.

```

.....      .....      .....      .....      .....      .....      .....
.X*.. -> ..*.. -> ..*.. -> ..*.. -> ..*.. -> ..*.. -> ..*..
.....      .X...      ..X..      ...X.      .....      .....      ...X.
..#..      ..#..      ..#..      ..#..      ..#X.      ..#X      ..#..
    
```

Pojašnjenje drugog test primjera: Ne uspijeva se izvršiti osmi potez (R) jer bi zmija izašla izvan matrice, zadnji potez se također zanemaruje. Prikazano je stanje igrice po potezima.

```

X...#      .X...#      .X...#      .....#      .....#      .....#      .....#      .....#
#*..#      #*..#      #X..#      #X..#      #..#      #..#      #..#      #..#
#...* -> #...* -> #...* -> #X.* -> #XX* -> #XXX. -> #XXX. -> #XX.
#*#*      #*#*      #*#*      #*#*      #*#*      #*#*      #*#X      #*#XX
#..#      #..#      #..#      #..#      #..#      #..#      #..#      #..#
#*..#      #*..#      #*..#      #*..#      #*..#      #*..#      #*..#      #*..#
    
```

Pojašnjenje trećeg test primjera: Ne uspijeva se izvršiti sedmi potez (D) jer bi zmija udarila u samu sebe, zadnji potez se također zanemaruje. Prikazano je stanje igrice po potezima.

```

..**      ..*X      ..XX      ..XX      ..XX      ..XX      ..X..
...X      ...X      ...X      ..X..      ..X..      ..X..      .XX..
.**.. -> .**.. -> .**.. -> .**.. -> .*X.. -> .XX.. -> .XX..
.....      .....      .....      .....      .....      .....      .....
.....      .....      .....      .....      .....      .....      .....
    
```