



23. siječnja 2012. od 14:30 do 16:30

Infokup 2012

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

Sadržaj

Zadaci.....	1
Zadatak: F1	2
Zadatak: Tenis.....	3
Zadatak: Restoran	4

Sponzori



Medijski pokrovitelji



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

udruga mladih programera
dump



Zadaci

U tablici možete pogledati ograničenja za zadatke:

Zadatak	F1	Tenis	Restoran
Vremensko ograničenje	10 sekunda	10 sekunda	10 sekunda
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 SmallBasicu 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.

Sretno i uspješno!



Zadatak: F1

40 bodova

Bernie E. nikad nije propustio utrku Svjetskog prvenstva u Formuli 1. Možda zato što jako voli gledati utrke brzih i skupih bolida, a možda i zato što je vlasnik tog „oktanskog cirkusa“. Tijekom utrke, Bernie E. promatra službeni monitor na kome se ispisuju trenutni rezultati utrke, tj. koliko trenutno koji bolid **vremenski zaostaje za vodećim** bolidom u utrci.

Međutim, njega više zanima **pozitivan vremenski razmak između bilo koja dva bolida**. Napiši program koji će omogućiti velikom gazdi da na osnovu službenih rezultata pronađe vremenski razmak koji ga zanima.

Ulaz

- proizvoljan niz realnih brojeva **X** ($0 \leq X \leq 59$) koji završava vrijednošću „0“ i pri čemu prvi učitani broj predstavlja zaostatak (u sekundama) zadnjeg bolida u poretku za trenutno vodećim bolidom, drugi učitani broj predstavlja zaostatak predzadnjeg u poretku za vodećim i tako sve do predzadnjeg učitano broj (zadnji je „0“) koji je zaostatak drugog u poretku za vodećim. U utrci će sudjelovati **najviše 15** bolida;
- prirodni brojevi **B1** i **B2** ($1 \leq B1 < B2 \leq 15$) jedan ispod drugog, trenutne pozicije bolida čiji vremenski razmak tražimo.

Izlaz

- pozitivan realan broj koji predstavlja traženi vremenski razmak.

Primjeri

	Ulaz	Izlaz	Objašnjenje
Primjer 1	34.23 29.56 18.15 4.23 2.06 0.34 0 2 4	3.89	Drugoplasirani bolid u utrci za vodećim zaostaje 0.34 sekunde, a četveroplasirani za vodećim zaostaje 4.23 sekunde. Vremenski razmak između promatrana dva bolida je 3.89 sekunda.



Zadatak: Tenis

70 bodova

Prvi potvrđeni spomen igre slične današnjem tenisu nalazimo u zapisima francuskih redovnika iz 11. i 12. stoljeća. Tenis kakav danas poznajemo je nastao oko 1874. kada je Englez Walter Wingfield definirao pravila igre i kada se 1877. po prvi put održao i planetarno popularni teniski turnir u Wimbledonu. Opišimo jedan dio pravila teniske igre koji ćemo koristiti u ovom zadatku.

U teniskom meču bodovanje se vrši po poenima, gemovima i setovima. Setovi se dijele na gemove, a gemovi na poene. Igrač je osvojio poen kada nakon razmjene udaraca loptice reketom, protivnički igrač više nije u stanju vratiti lopticu u tenisko polje na pravilan način.

Gem započinje rezultatom „0:0“ u poenima. Nakon određenog broja odigranih poena jedan od igrača osvaja gem. To uspijeva onom tko osvoji **najmanje četiri poena i ima barem dva poena više od protivnika**. U gemu postoji posebno označavanje osvojenih poena. Prvi osvojeni poen u gemu se za igrača označava s 15, drugi s 30, a treći s 40.

Kada pri rezultatu „40:40“ (tj. svaki od igrača je osvojio po tri poena) neki od igrača osvoji poen, tada on dolazi u prednost (u oznaci **AD-oznaka_igrača**). Osvoji li igrač u prednosti još jedan poen tada osvaja i cijeli gem. Ako protivnik koji nije u prednosti osvoji poen, tada se proglašava izjednačenje (u oznaci „**deuce**“). Igra se nastavlja sve dok jedan od igrača ne osvoji dva uzastopna poena te tako i cijeli gem.

Napiši program koji za **trenutni rezultat u gemovima** nekog trening meča dvojice igrača i **opis prvih N odigranih poena u sljedećem gemu** koji će igrati, određuje i ispisuje novi rezultat u gemovima i poenima.

Ulaz

- prirodni brojevi **AG** i **BG** ($0 \leq \text{AG}, \text{BG} \leq 6$) jedan ispod drugog, trenutni broj osvojenih gemova igrača s oznakom „A“ i igrača s oznakom „B“;
- prirodan broj **N** ($1 \leq \text{N} \leq 20$), broj odigranih poena koje opisujemo;
- u sljedećih **N** redaka se nalazi po jedan znak („A“ ili „B“) koji predstavlja oznaku igrača koji je osvojio i-ti poen.

Izlaz

- u jedini redak izlaza treba ispisati traženi rezultat u obliku „**AG:BG AP:BP**“ pri čemu će uvijek vrijediti da je $0 \leq \text{AG}, \text{BG} \leq 7$, a **AP** i **BP** mogu biti prirodni brojevi iz skupa {0, 15, 30, 40} ili tekst „AD-A“ ili „AD-B“ ili „deuce“.

Primjeri

Primjer 1		Primjer 2		Primjer 3	
Ulaz	Izlaz	Ulaz	Izlaz	Ulaz	Izlaz
2 1 3 A A B	2:1 30:15	1 3 6 A A B A B A	2:3 0:0	4 0 9 A A B A B B A B B	4:0 AD-B



Zadatak: Restoran

90 bodova

Klještić je vlasnik „Rakove poslastice“, najpoznatijeg ugostiteljskog objekta pod morem. U posljednje vrijeme sve manje posjetitelja dolazi u njegov restoran na daleko poznati Rakburger. Kako bi povećao zaradu, vlasnik je odlučio tijekom vikenda provoditi neke zanimljive akcije.

Tako je obećao da će **svake subote jedan cijeli sat** (koji počinje na cijeli broj, npr. od 10:00 do 11:00) unutar radnog vremena restorana proglasiti „**Sretnim satom**“ te da će svim posjetiteljima restorana koji su unutar tog sata kupili Rakburger sljedeći dan vratiti potrošeni novac.

Isto je tako obećao da će **svake nedjelje nekih 15 uzastopnih minuta** unutar radnog vremena restorana (npr. od 14:34 do 14:49) proglasiti „**Sretnom četvrtinom**“ te da će svim posjetiteljima restorana koji su u tih 15 minuta kupili Rakburger sljedeći dan vratiti potrošeni novac.

Klještić želi pažljivo odabrati „Sretni sat“ i „Sretnu četvrtinu“ (ovisno o danu). Zato po isteku radnog vremena **sakupi sve račune** izdane tog dana i odabere onaj sat/četvrtinu unutar radnog vremena u kome je ukupna vrijednost naplaćenih računa bila **najmanja i različita od nule**. Ako znamo da je radno vrijeme „Rakove poslastice“ od **9 sati u jutro do 15 sati popodne**, pomogni Klještiću da odabere sat/četvrtinu koji će proglasiti sretnim/sretnom. Ako postoji više takvih mogućnosti, tada on odabire onu koja je prije započela.

Ulaz

- tekst „subota“ ili „nedjelja“, koji označava dan u tjednu kada promatramo račune;
- prirodan broj N ($1 \leq N \leq 100$), broj računa izdanih tijekom radnog vremena;
- u sljedećih N redaka se nalaze po 3 prirodna broja oblika „**S M C**“ pri čemu je **S** ($9 \leq S \leq 14$) sat i **M** ($0 \leq M \leq 59$) minuta izdavanja i-tog računa, a **C** ($1 \leq C \leq 100$) vrijednost potrošenog novca koja piše na i-tom računu.

Izlaz

- tekst oblika „**SP:MP-SK:MK**“, pri čemu su SP i MP sat i minuta **od kada počinje** sretni sat/četvrtina, a SK i MK sat i minuta **do kada traju** sretni sat/četvrtina (ta minuta nije uključena u „Sretni sat/četvrtinu“). Dovoljena su dva oblika ispisa („09:05-09:20“ ili „9:5-9:20“).

Napomena: „subota“ će se pojaviti u 70% službenih test primjera

Primjeri

	Ulaz	Izlaz	Objašnjenje
Primjer 1	subota 5 9 15 17 10 47 15 13 33 20 14 21 54 12 49 32	10:00-11:00	– od 9:00 do 10:00 je potrošeno 17 – od 10:00 do 11:00 je potrošeno 15 – od 11:00 do 12:00 je potrošeno 0 – od 12:00 do 13:00 je potrošeno 32 – od 13:00 do 14:00 je potrošeno 20 – od 14:00 do 15:00 je potrošeno 54
Primjer 2	nedjelja 5 9 15 17 10 47 15 13 33 20 14 21 54 12 49 32	10:33-10:48	