

Univerzitet u Sarajevu
Elektrotehnički fakultet u Sarajevu

KANTONALNO TAKMIČENJE IZ INFORMATIKE
ZA OSNOVNE ŠKOLE
ZADACI ZA PRIPREMU

Sarajevo, Februar 2012. godine

Sadržaj:

Pregled oblasti i tema za Općinsko takmičenje iz Informatike učenika osnovnih škola Kantona Sarajevo....	3
1.Zadaci sa općinskih takmičenja.....	4
Općinsko takmičenje iz informatike 2010. godine.....	4
2.Zadaci sa kantonalnih takmičenja.....	7
XII Kantonalno takmičenje iz informatike (2008. godine).....	7
XIII Kantonalno takmičenje iz informatike (2009. godine).....	8
XIV Kantonalno takmičenje iz informatike (2010. godine).....	10
XV Kantonalno takmičenje iz informatike (2011. godine).....	12
3.Zadaci sa državnih takmičenja.....	15
1. Juniorsko takmičenje mladih informatičara BiH (2008.).....	15
3. Državno takmičenje iz informatike za osnovne škole (2009.).....	19
4. Državno takmičenje iz informatike za osnovne škole (2010.).....	23

Pregled oblasti i tema za Općinsko takmičenje iz Informatike učenika osnovnih škola Kantona Sarajevo

Jednostavni ulaz i izlaz

Uslovi i grananje

Petlje

Strukturna dekompozicija problema (razbijanje problema na podprobleme)

Jednostavne matematske operacije

Nalaženje maksimuma i minimuma za zadate vrijednosti

Operacije cjelobrojnog dijeljenja

Korištenje ostatka pri cjelobrojnog dijeljenju

Razne operacije (sumiranje, brojanje, itd.) nad ulaznim vrijednostima po zadatim uslovima

Ispisivanje na ekran po zadatim uslovima

Učitavanje podataka s tastature po zadatim uslovima

Nizovi

Učitavanje i ispisivanje elemenata niza

Pretraživanje nizova

Nalaženje sume elemenata, najmanjeg i najvećeg elementa i sl.

Stringovi

Učitavanje i ispisivanje stringova

Razne analize stringova

Pretraživanje stringova

Brojanje određenih karaktera u stringovima

Izdvajanje podstringova

Konverzije podataka

Karakter u broj (broj u karakter)

String u broj (broj u string)

Tekstualne datoteke

Čitanje iz tekstualni datoteka

Pisanje u tekstualne datoteke

Ispisivanje podataka u datoteku pod zadatim uslovom

Sortiranje

1. Zadaci sa općinskih takmičenja

Općinsko takmičenje iz informatike 2010. godine

Zadatak 1: Minimalni broj

Sa standardnog ulaza učitati tri cijela broja. Na standardni izlaz odštampati najmanji broj.

Primjer 1:

5
3
9
Najmanji broj je 3

Primjer 2:

4
8
4
Najmanji broj je 4

Zadatak 2: Parni

Sa standardnog ulaza učitati skup cijelih brojeva, kraj unosa 0. U datoteku Parni.dat, na direktoriju c:\OT2010\ odštampati pozitivne parne brojeve.

Primjer:

Unesi skup brojeva (0 kraj unosa):

1
2
4
-6
7
9
11
0

Lista parnih brojeva:

2
4

Zadatak 3: Trougao

Sa standardnog ulaza učitati dimenzije trouglova, kraj unosa jedna od unijetih dimenzija 0. Nakon svakog unosa na standardni izlaz odštampati date dimenzije i ispisati

- ako je dati trougao jednakostanični u formatu :

“Trougao XXX – YYY - ZZZ je jednakostranični”

- ako je dati trougao jednakokraki:

“Trougao XXX – YYY - ZZZ je jednakokraki”

- inače
"Trougao XXX – YYY - ZZZ nije jednakokraki"
- unijeta vrijednost 0:
"Kraj rada"

XXX – prva dimenzija trougla (tri mjesta za ispis)
 YYY – druga dimenzija trougla (tri mjesta za ispis)
 ZZZ – treća dimenzija trougla (tri mjesta za ispis)

(Jednakokraki trougao je onaj trougao kod koga su dvije stranice (kraka) jednake a jednakostranični trougao je onaj trougao kod koga su sve stranice jednake. Nije potrebno provjeravati da li unesene vrijednosti predstavljaju validne stranice nekog trougla.)

Primjer:

Unesi dimenzije trougla: 15, 13, 140
 Trougao 15 - 13 - 140 nije jednakokraki

Unesi dimenzije trougla: 150, 105, 105
 Trougao 150 - 105- 105 je jednakokraki

Unesi dimenzije trougla: 10, 10, 10
 Trougao 10 - 10 - 10 je jednakostranični

Unesi dimenzije trougla: 15, 0, 140
 Kraj rada

Napomena: Ako se koristi Pascal onda se izostavlja zarez kao separator između dimenzija trougla.

Zadatak 4: Naopako

Sa standardnog ulaza učitati rečenicu. Na standardni izlaz i u datoteku Naopako.dat, na direktoriju c:\OT2010\ odštampati rečenicu naopake.

Primjer:

Unesi rečenicu:Opcinsko takmicenje iz informatike
 Naopacke: ekitamrofni zi ejnecimkat oksnicpO

Zadatak 5: Riječnik

Iz datoteke Ulaz.dat, na direktoriju c:\OT2010\ pročitati tekst. Sa standardnog ulaza učitati riječi:

- riječ koja je mijenja
- riječ sa kojom se mijenja.

Na standardni izlaz odštampati učitani tekst vršeći zamjenu riječi koja se mijenja sa riječju kojom se mijenja. Riječ predstavlja skup malih i velikih slova engleskog alfabeta, svi ostali znakovi (razmak, tačka, cifre itd) predstavljaju znakove interpukcije (razdvajaju riječi).

Primjer:

U datoteci Ulaz.dat se nalazi slijedeći tekst:

Ovo je, tekst.m. za probu.

-Ima čak dvije linije teksta, pa da provjerimo kako radi

Unesi riječ koja se mijenja: tekst

Unesi riječ kojom se mijenja: test

Na standardni izlaz treba ispisati:

Ovo je, test.m. za probu.

-Ima čak dvije linije teksta, pa da provjerimo kako radi

2. Zadaci sa kantonalnih takmičenja

XII Kantonalno takmičenje iz informatike (2008. godine)

Zadatak 1.

U datoteci «Imena.dat» data su imena poznatih ličnosti 19. i 20. vijeka. U datoteci «Prezimeni.dat» data su njihova prezimena u različitom rasporedu u odnosu na datoteku «Imena.dat». Datoteka «Komplet.dat» sadrži imena i prezimena tih ličnosti u cjelosti u formatu: ime, blanko, prezime.

Napisati program koji uparuje imena i prezimena iz datoteka «Imena.dat» i «Prezimeni.dat» i u izlaznu datoteku «Rezultat.dat» upisuje ime, prezime, broj reda imena iz ulazne datoteke «Imena.dat», broj reda prezimena iz ulazne datoteke «Prezimeni.dat».

Zadatak 2.

Sa standardnog ulaza učitati iznose novčanica u kasi. Moguće novčanice su 50, 20, 10, 5, 2, 1 KM i 5, 10, 20 i 50 feninga.

Prodavac treba da vrati kusur tako da upotrijebi najmanje broj novčanica. Sa standardnog ulaza čitati iznos računa. Plaćanje se vrši u novčanicama od 100 KM. Za dati kusur ispisati koliko kojih novčanica treba vratiti. Kraj programa je unijeti iznos računa 0 KM ili kada nije moguće vratiti kusur, uz adekvatnu poruku.

Zadatak 3.

U datoteci «Gradovi.dat» data su imena gradova. Napisati program koji čita datu datoteku i u datoteku «Izlaz.dat» upisuje dati skup sortiran od manjeg ka većem izbacujući višestruko unijete.

XIII Kantonalno takmičenje iz informatike (2009. godine)

Zadatak 1: Podmornice

Datoteka "Ulaz.txt" na direktoriju "C: \KT2009\ZADATAK1\" sadrži deset redova po deset cifara 0 i 1 u svakom redu. "1" predstavlja "podmornicu", koje mogu sadržavati:

- 1 polje 4 podmornice,
- 2 polja 3 podmornice,
- 3 polja 2 podmornice,
- 4 polja 1 podmornica.

Polja kod "podmornica" koje su složena vezana su gore, dolje, lijevo i desno.

Napisati program koji čita datu datoteku i korisniku pruža mogućnost pogađanja pozicije "podmornica", naslijedeći način:

- korisnik unosi poziciju (od 1-10 i od A – J),
- nije pogođena podmornica program ispisuje (nije pogođen)
- pogođeno polje podmornice
 - o podmornica potopljena (Potopljena podmornica)
 - o podmornica nije potopljena (Pogodak)

Kraj igre kada su sve podmornice potopljene, ili ako korisnik unese poziciju 0.

Na kraju program štampa broj odigranih poteza.

Zadatak 2: Broj

Datoteka „Broj.txt“ na direktoriju "C: \KT2009\ZADATAK1\" sadrži 15 različitih cijelih brojeva koji mogu i da se ponavljaju. Napisati program koji treba da ispiše učitane brojeve, ali tako da se višestruka pojavljivanja jednog broja ne ispisuju posebno, već da se svaki broj napiše tačno jednom, a uz njega i na kojim mjestima u ulaznoj datoteci se pojavio.

Primjer:

Brojka 17 je na mjestima 1, 7, 12, 13.

Brojka 3 je na mjestima 2, 3, 4, 22, 39.

Zadatak 3: Pogađanje

Datoteka „Rijec.txt“ na direktoriju "C: \KT2009\ZADATAK1\" sadrži tri riječi koje je potrebno pogoditi. Ako korisnik pritisne slovo koje je u riječi dobiva broj bodova jednak poziciji slova u riječi pomnožen sa brojem linije u kojoj se riječ nalazi. Ako riječ ima više istih slova i ako više riječi ima ista slova tada se dobivaju bodovi za svako slovo u riječi i za svaku riječ. Takođe ispisuje date riječi tako da za slova kaja nisu pogođena ispisuje „-„.

Nakon uspješno pogođene riječi ili nakon pet pokušaja pogađanja bez bodova, završiti program ispisivanjem na ekran riječi koje su se pogađale, broj osvojenih bodova po svakoj riječi i otkrivena slova svake riječi.

Razigravanje: Put

Dva robota krenu iz iste tačke. Put koji pređu svake minute definisan je u tabelama „Put1.txt“ i „Put2.txt“ direktoriju "C: \KT2009\Razig\" i to u formi:

X L (X: L - lijevo, R – desno, U – gore, D – dole), (L: cijeli broj, dužina u metrima).

Napisati program koji čita date datoteke i ispisuje u kojoj minuti će se dati roboti sresti u istoj tački i poziciju te tačke u odnosu na početak, u formatu:

8 minuta, L – 50, U – 30.

XIV Kantonalno takmičenje iz informatike (2010. godine)

Zadatak 1: Put

Dva robota krenu iz iste tačke. Put koji pređu svake minute definisan je u tabelama „Put1.txt“ i „Put2.txt“ direktoriju “C: \KT2010\Zadatak1\” i to u formi:

X L (X: L - lijevo, R – desno, U – gore, D – dole), (L: cijeli broj, dužina u metrima).

Napisati program koji čita date datoteke i ispisuje u kojoj minuti će se dati roboti sresti u istoj tački i poziciju te tačke u odnosu na početak, u formatu:

8 minuta, L – 50, U – 30.

Zadatak 2: Tetris

U datoteci Tetris.dat, na direktoriju C: \KT2010\Zadatak2\ dat je skup parova u obliku:

$x - y$

Svaki par se nalazi u zasebnom redu. x,y su brojevi koji predstavljaju slijedeće:

x – redni broj bloka prema tabeli:

1	2	3	4	5	6	7
*	**	*	**	***	*	***
		*	**		*	***
					*	***

y - kolona u kojoj se nalazi gornji lijevi ugao bloka

Blokovi se smještaju u matricu dimenzija 10x10 i to tako da kreću od vrha matrice, a zatim se spuštaju prema dolje sve dok ne naiđu na dno matrice ili je jedan od elemenata matrice već popunjen blokom.

Potrebno je na ekran ispisati stanje matrice nakon završetka učitavanja datoteke i to tako da su popunjeni elementi matrice označeni slovom X, a nepopunjeni znakom minus (-).

Pri tome je garantovano da blokovi specificirani u datoteci Tetris.dat neće prekoračiti dimenzije matrice po širini niti po visini.

Primjer izlaza:

```
-----
-----
-----
-----XX
-----XXX-
-----XXX-
-----XXX-
-XX- - -XX-
-XX-X- -XX-
--XXX- -X-X
```

Zadatak 3: Cifre

Sa standardnog ulaza učitati skup znakova maksimalne dužine 50. Na standardni izlaz i u datoteku Cifre.dat, na direktoriju C:\KT2010\Zadatak3 odštampati dati skup sortiran od većeg ka manjem (neopadajući), izbacujući višestruko unijete cifre i sve što nije cifra i sumu datih cifara u formatu:

Rezultat: XXXXXXXXXXXXXXXXXXXXXXXX
Suma: YYYYY

Primjer:
Unesi skup cifara: 345004567A90239
Rezultat: 9765430
Suma: 34

Razigravanje: Rečenice

Data je datoteka "Text.dat", na direktoriju C:\KT2010\Razig\, koja sadrži skup rečenica. Svaka rečenica završava sa ":", "!" ili "?" i može biti u više redova. Maksimalni broj rečenica je 50, a maksimalna dužina rečenica može biti do 400 znakova. U datoteku "Sort.dat", na direktoriju C:\KT2010\Razig\ ispisati dati skup rečenica sortiran po dužini rečenice, svaka rečenica se nalazi u jednom redu.

XV Kantonalno takmičenje iz informatike (2011. godine)

Zadatak 1: Igra pogađanja brojeva

Potrebno je napisati program koji pogađa cijeli broj između 1 i 100 (uključujući i ta dva) koji je zamislio korisnik.

Program najprije treba ispisati poruku “Zamislite jedan cijeli broj između jedan i stotinu”, a zatim treba ispisivati poruke tipa “Da li je zamišljeni broj 50?” na koje korisnik treba odgovoriti:

- slovom V ako je zamislio broj veći od ponuđenog
- slovom M ako je zamislio broj manji od ponuđenog
- slovom D ako je zamislio ponuđeni broj
- slovom I ako želi prekinuti igru i izaći iz programa.

Ako je odgovor D ili I igra se prekida uz odgovarajuću poruku “Pogodio sam iz X pokušaja!” (X predstavlja broj pokušaja) odnosno “Igra je prekinuta!”, a ako je odgovor V ili M program treba pokušati neki drugi broj.

Prilikom pogađanja brojeva, program treba koristiti metodu polovljenja intervala (iz intervala mogućih tačnih rješenja, za sljedeći pokušaj treba odabrati vrijednost na sredini intervala). Program treba biti u mogućnosti da iz maksimalno 7 pokušaja pogodi zamišljeni broj.

Primjer:

Zamislite jedan cijeli broj između jedan i stotinu

Da li je zamišljeni broj 50?

M

Da li je zamišljeni broj 25?

M

Da li je zamišljeni broj 12?

V

Da li je zamišljeni broj 18?

V

Da li je zamišljeni broj 21?

V

Da li je zamišljeni broj 23?

M

Da li je zamišljeni broj 22?

D

Pogodio sam iz 7 pokušaja

!

Zadatak 2: Matematički izraz

Napisati program koji iz tekstualne datoteke PROBLEMI.TXT, na direktoriju C: \KT2011\Zadatak2 učitava niz jednostavnih matematičkih izraza, a u tekstualnu datoteku RJESENJA.TXT upisuje rješenja tih izraza.

Svi izrazi u ulaznoj datoteci su oblika

OPERAND OPERATOR OPERAND

pri čemu su oba OPERANDa cijeli brojevi, a OPERATOR može biti znak: + (sabiranje) - (oduzimanje) * (množenje) ili / (dijeljenje). Između operatora i operanda nema razmaka. Izrazi su međusobno razdvojeni novim redom (svaki izraz se nalazi u zasebnom redu datoteke).

U izlaznu datoteku treba upisati rješenja matematičkih operacija sljedećeg oblika

OPERAND OPERATOR OPERAND = REZULTAT

OPERATOR i OPERAND su kao i ranije, a nakon njih treba upisati znak = i realan broj REZULTAT koji predstavlja rezultat matematičke operacije koju čine operandi i operator. Kao i u ulaznoj datoteci, između različitih elemenata rješenja nema razmaka, a rješenja trebaju biti međusobno razdvojena novim redom.

Primjer:

PROBLEMI.TXT

2+2

145*2

567-123

RJESENJA.TXT

2+2=4

145*2=290

567-123=444

Zadatak 3: Sutrašnji datum

Data je datoteka „DATUMI.TXT“, koja sadrži datume u formatu:

dd mm gggg

Napisati program koji nalazi predhodni i naredni datum i ispisuje u izlaznu datoteku „REDDAT.TXT“ u formatu

Datum: dd.mm.gggg. Prethodni dd.mm.gggg. Naredni dd.mm.gggg. godine

Voditi računa o prestupnim godinama. Koristite sljedeće pravilo:

Da bi neka godina bila prestupna mora biti djeljiva sa 4. Međutim, od onih godina koje su djeljive sa 4 nisu prestupne one koje su djeljive sa 100 a nisu sa 400. Npr. 1900 nije bila prestupna dok je 2000 godina bila prestupna. U prestupnoj godini mjesec februar ima 29 dana dok u ostalim godina ima 28.

Primjer:

DATUMI.TXT

12 04 2010

28 02 2009

28 02 2008

REDDAT.TXT

Datum: 12.04.2010. Prethodni 11.04.2010. Naredni 13.04.2010. godine

Datum: 28.02.2009. Prethodni 27.02.2009. Naredni 01.03.2009. godine

Datum: 28.02.2008. Prethodni 27.02.2008. Naredni 29.02.2008. godine

Zadatak 4: Rastući podnizovi

Tekstulna datoteka Cifre.txt sadrži skup cifara od 0 do 9, bez razmaka, ostalih karaktera i znakova interpunkcije. Napisati program koji čita datu datoteku i u izlaznu tekstualnu datoteku Nizovi.txt izdvaja i upisuje strogo rastuće podnizove iz datog skupa cifara. Svaki podniz upisati u novi red. Niz se sastoji od najmanje dvije cifre.

Primjer 1:

Strogo rastući podniz je:

12389,

a nizovi:

4

44

963

nisu strogo rastući podnizovi

Primjer 2:

Za skup cifara:

12344456542890102963

Rezultat:

1234

456

289

01

02

3. Zadaci sa državnih takmičenja

1. Juniorsko takmičenje mladih informatičara BiH (2008.)

Zadatak 1: broj

Dozvoljeno vrijeme izvršavanja: 2s

Dozvoljeno korištenje memorije: 32MB

Sa standardnog ulaza potrebno je pročitati broj u opsegu od 2 do 30000. Za dati broj, na standardnom izlazu, potrebno je ispisati sve različite proste faktore tog broja u opadajućem redoslijedu. Ispis treba biti u jednom redu. Prosti faktori broja su prosti oni brojevi, s kojima je dati broj djeljiv. Prosti broj je svaki broj koji je djeljiv samo sa jedan i sa samim sobom, s tim da 1 nije prost broj.

Primjer: faktori broja 30 su 15, 10, 6, 5, 3, 2; dok su prosti faktori samo 5, 3 i 2.

Primjer 1

Ulaz:

15

Izlaz:

5 3

Primjer 2

Ulaz:

3000

Izlaz:

5 3 2

Izvršnu datoteku broj.exe smjestiti u direktorij C:\JBHOI\

Zadatak 2: grafika

Dozvoljeno vrijeme izvršavanja: 2s

Dozvoljeno korištenje memorije: 32MB

Sa standardnog ulaza potrebno je učitati niz, pozitivnih cijelih brojeva manjih od 20. Niz brojeva se završava brojem 0, koji se ne uzima u obzir prilikom dole navedenog zadatka. Niz neće sadržavati više od 80 brojeva. Svi brojevi se nalaze u jednom redu.

Svaki od brojeva predstavlja vrijednost koju treba prikazati na grafikonu. Grafikon se crta pomoću znakova 'x', tako da se ispiše onoliko x-ova okomito koliko odgovara broju zadanom na ulaz. Na izlazu ne smije biti praznih redova. Na kraju svakog reda mora biti znak x odmah prije znaka za novi red.

Primjer:

Radi vidljivosti, razmak u primjeru je predstavljen znakom ' '.

XX . . XX . XX

XX . XXX . XX

XXXXXXXXXXXXXX

Korektno

XX . . XX . XX . . .

XX . XXX . XX . . .

XXXXXXXXXXXXX

Nekorektno

Zabranjena je upotreba funkcija za pomjeranje kursora po ekranu.

Primjer 1

Ulaz:

5 6 1 2 7 9 1 7 9 1 3 4 0

Izlaz:

```

      X  X
      X  X
     XX XX
    X  XX XX
   XX  XX XX
  XX  XX XX  X
 XX  XX XX XX
XX XXX XX XX
XXXXXXXXXXXXX
```

Primjer 2

Ulaz:

1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 0

Izlaz:

```

      X
     XXX
    XXXXX
   XXXXXXX
  XXXXXXXXX
 XXXXXXXXXXX
XXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
```

Izvršnu datoteku grafika.exe smjestiti u direktorij C:\JBHOI\

Zadatak 3: rijec

Dozvoljeno vrijeme izvršavanja: 2s

Dozvoljeno korištenje memorije: 32MB

Sa standardnog ulaza potrebno je učitati nekoliko riječi. Svaka riječ se nalazi u zasebnom redu i sastoji se samo od malih slova engleske abecede, dužine do 100 znakova. Na ulazu može biti najviše 100

riječi. Riječ koja označava kraj učitavanja i ne uzima se u obzir prilikom traženja najtežih riječi (definirano u nastavku) je „zadnja“. Slova nj, lj i dz tretirati kao dva odvojena slova. Za potrebe ovog zadatka, samoglasnici su sljedeći: a, e, i, o, u.

Potrebno je na standardni izlaz ispisati drugu po redu najtežu riječ za pročitati. Težina čitanja riječi je definirana brojem parova susjednih suglasnika u toj riječi. U slučaju pojave više od dva suglasnika uzastopno u riječi, svaka dva susjedna suglasnika povećavaju težinu za jedan.

Riječ	Težina
ja	0
olovo	0
skok	1
stvar	2
stvarnost	4

Rješenje će uvijek biti jedinstveno.

Primjer 1

Ulaz:

ja
olovo
skok
stvar
stvarnost
zadnja

Izlaz:

stvar

Primjer 2

Ulaz:

otorinolaringologija
nabukodonosor
triangl
paprika
neslužbena
zadnja

Izlaz:

neslužbena

Izvršnu datoteku rijec.exe smjestiti u direktorij C:\JBHOI\

3. Državno takmičenje iz informatike za osnovne škole (2009.)

Zadatak 1: cifre

Data su vam dva velika broja, gdje svaki može imati do 10000 cifara. Oba broja imaju isti broj cifara. Vaš zadatak je da odredite koliko najmanje cifara trebate promijeniti u prvom broju tako da se ta dva broja sadrže od istih cifara.

Ulažni podaci

U prvoj liniji na standardnom ulazu se nalazi jedan cio broj *a* (dužine do 10000 cifara). U drugoj liniji se nalazi drugi broj, broj *b*, koji ima isti broj cifara kao broj *a*.

Izlazni podaci

Na prvoj liniji i jedinoj liniji standardnog izlaza potrebno je ispisati najmanji broj cifara koje treba promijeniti u broju *a* tako da se brojevi *a* i *b* sadrže od istih cifara.

Primjer 1

cifre.in

51

16

cifre.out

1

Objašnjenje prvog primjera

Ako cifru 5 iz prve cifre zamijenimo sa 6, dobijemo 61, a brojevi 61 i 16 se sastoje od istih cifara.

Primjer 2

dnk.in

661

116

cifre.out

1

Objašnjenje drugog primjera

Ako cifru 6 iz prvog broja zamijenimo sa 1, dobijemo 611, a brojevi 611 i 116 se sastoje od istih cifara. To je najmanji broj promjena.

Ograničenja na resurse

Vaš program se treba izvršavati za ne više od 1s i ne smije koristiti više od 32 MiB memorije po svakom testnom slučaju.

Zadatak 2: Prenos

U nižim razredima osnovne škole ste učili kako sabrati dva velika prirodna broja metodom potpisivanja. Saberimo brojeve 543 i 695:

$$\begin{array}{r}
 543 \\
 + 695 \\
 \hline
 1238
 \end{array}$$

Da se prisjetimo: krenemo s desna na lijevo, te sabiramo brojeve koji su potpisani. Prvo saberemo 3 i 5, što kao rezultat daje 8. Zatim saberemo 4 i 9, što kao rezultat daje 13. U tom slučaju pišemo cifru jedinica, odnosno 3, a cifru desetica pamtimo i prenosimo. Cifru desetica zovemo i *prenos*. Sada sabiramo 5 i 6, te na to dodajemo prenos, odnosno 1. Kao rezultat dobijamo 12. Ponovo pišemo 2, te pamtimo 1 kao prenos. Ako više nemamo šta sabrati, tada samo na početak lijepimo prenos. U ovom slučaju samo dodamo 1.

Tokom ovog sabiranja smo izvršili dvije operacije prenosa. Vaš zadatak je da za dva unesena broja a i b pronađete koliko puta trebamo izvršiti prenos da bismo sabrali ta dva broja metodom potpisivanja.

Ulazni i izlazni podaci

U ulaznoj datoteci prenos.in se nalaze dva prirodna broja a i b. U izlaznu datoteku prenos.out trebate samo ispisati broj operacija prenosa koje su se desile prilikom sabiranja ova dva broja. Pretpostavite da brojevi a i b nemaju više od 9 cifara.

Primjeri

prenos.in	prenos.out
10 18	0
356 669	3
5 9	1

Zadatak 3: tablica

Za današnji svijet se kaže da je zasnovan na protoku informacija. Svakog časa se razmijene ogromne količine informacija. To sigurno dovodi do velikog nereda, te se iz časa u čas sve teže snaći među svim tim različitim informacijama. Zbog toga se smišljaju raznorazne metode kako te informacije prikazati na bolji (čitljiviji, jasniji) način. Jedna od tih strategija je poznata pod nazivom *data mining*.

Od vas se ipak ne očekuje da radite takve kompleksne stvari kao što je data mining (barem ne još). Vama su data dva prirodna broja, a i b. Vaš je zadatak da za svaki broj između a i b (uključujući i a i b) ispišete taj broj, njegov kvadrat i njegov kub, ali na pregledan način radi što lakše obrade datih podataka. Način ispisivanja ćemo detaljnije objasniti.

Na primjer, neka je a = 2 i b = 6. Tablica bi trebala izgledati ovako:

+	+	+	+	+	+	+	+	+	+
	2		4		8				
+	+	+	+	+	+	+	+	+	+
	3		9		27				
+	+	+	+	+	+	+	+	+	+
	4		16		64				
+	+	+	+	+	+	+	+	+	+
	5		25		125				

```

+---+---+---+
| 6 | 36 | 216 |
+---+---+---+

```

Svaka kolona treba biti dugačka onoliko koliko je dugačak najduži broj u koloni. Računajte i na to da prije i poslije tog broja mora stajati razmak. Dvije različite kolone ne moraju nužno biti iste dužine. Također se morate i pobrinuti za ispis okvira, koji mora biti isti kao u gornjem primjeru. Na kraju reda ili kolone NE SMIJE biti viška razmaka (kao ni unutar tabele).

Ulazni podaci

U prvoj liniji ulazne datoteke „tablica.in“ se nalazi broj a , a u drugoj liniji broj b , pri čemu je $1 \leq a \leq b \leq 100$. Dati brojevi su prirodni brojevi.

Izlazni podaci

U izlaznoj datoteci „tablica.out“ potrebno je ispisati tablicu na način kako je opisano u tekstu.

Primjer 1

tablica.in

1 4

tablica.out

```

+---+---+---+
| 1 | 1 | 1 |
+---+---+---+
| 2 | 4 | 8 |
+---+---+---+
| 3 | 9 | 27 |
+---+---+---+
| 4 | 16 | 64 |
+---+---+---+

```

Primjer 2

tablica.in

18 20

tablica.out

```

+---+---+---+
| 18 | 324 | 5832 |
+---+---+---+
| 19 | 361 | 6859 |
+---+---+---+
| 20 | 400 | 8000 |
+---+---+---+

```

4. Državno takmičenje iz informatike za osnovne škole (2010.)

Zadatak 1: BiH

Nekada davno, ljudi su koristili bušene kartice za čuvanje i manipulaciju podacima. Podaci su zapisivani pomoću tačkica, probušenih ili ne, pravilno raspoređenih na zamiljenoj mreži koja se nalazila na manjem ili većem kartonu. Različiti modeli su korišteni za zapisivanje slova i cifara na takvim karticama, neki jako efikasni a neki samo interesantni. Mi ćemo ovdje razmotriti sistem koji se još uvijek može vidjeti gdje tačkicama „crtamo“ slova. Na primjer slovo B, koristeći nule i jedinice umjesto bušenih i nebušenih tačkica se može zapisati kao

```
1 1 0
1 0 1
1 1 0
1 0 1
1 1 0
```

Na sličan način slovo H možemo zapisati kao

```
1 0 1
1 0 1
1 1 1
1 0 1
1 0 1
```

Jedna kartica je data kao mreža ili matrica nula i jedinica. Na primjer mreža dimenzija 9x16 je data sa

```
0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0
0 1 0 1 1 1 0 1 0 1 1 1 1 1 1 0
0 0 0 0 0 1 1 0 0 1 1 1 1 1 1 0
1 0 0 0 1 1 0 1 0 1 1 1 1 1 1 0
1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 0
1 0 0 0 1 1 0 1 0 1 1 1 1 0 1 0
1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0
1 0 0 0 1 1 0 1 0 1 1 1 1 0 1 0
1 1 1 1 1 1 1 0 0 0 0 0 1 0 1 0
```

U prvom redu na šestom mjestu možemo propoznati slovo B dok u 5 redu na poziciji 13 možemo vidjeti slovo H. To su jedina slova B i H na ovoj kartici. Vaš zadatak je da za datu karticu izbrojiti ukupan broj pojavljivanja slova B i ukupan broj pojavljivanja slova H i izvijestite o njihovom zbiru.

Ulazni podaci

Ulazna datoteka je bih.in. Prva linija ulazne datoteke sadrži dva cijela broja, M i N ($1 \leq M, N \leq 1000$) gdje M označava ukupan broj redova a N ukupan broj nula i jedinica u jednom redu date mreže. U sljedećih M linija dati su nizovi nula i jedinica koji predstavljaju karticu na opisani način.

Izlazni podaci

Izlazna datoteka je bih.out. Sadrži samo jednu liniju sa jednim brojem, ukupnim brojem slova B i slova H u mreži.

Primjer 1

Input(bih.in)

9 16

0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0
0	1	0	1	1	1	0	1	0	1	1	1	1	1	1	0
0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	0
1	0	0	0	1	1	0	1	0	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0	0	0	0	0	1	0	1	0
1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0
1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0
1	0	0	0	1	1	0	1	0	1	1	1	1	0	1	0
1	1	1	1	1	1	1	0	0	0	0	0	1	0	1	0

Output(bih.out)

4

Primjer 2

Input(bih.in)

5 8

0	1	1	1	1	1	1	0
0	1	0	1	1	1	0	1
0	0	0	0	0	1	1	1
1	0	0	0	1	1	0	1
1	1	1	1	1	1	1	0

Output(bih.out)

0

Zadatak 2: Energija

Proizvodnja i optimalno upravljanje proizvodnjom energije je od velike važnosti za svako društvo. Procjenjuje se da će Bosna i Hercegovina za deset godina imati do 1000 različitih postrojenja za proizvodnju električne energije. Pored postojećih termo i hidro elektrana izgrađiće se cijeli parkovi novih malih hidro i vjetro elektrana. Svaka od elektrana može raditi sa punom ali i sa svakom drugom manjom snagom većom od nula. Električna energija se teško skladišti pa je potrebno proizvesti tačno onoliko energija kolika je potražnja za nju.

Vaš zadatak je da napišete program koji će za datu potražnju za energijom u jednom danu i za datu listu elektrana sa njihovom maksimalnom proizvodnjom i i cijenom proizvodnje po jedinici naći najbolji (najjeftiniji plan proizvodnje).

Ulazni podaci

Ulazna datoteka je energija.in. Prva linija ulazne datoteke sadrži broj N ($1 \leq N \leq 1000$) koji označava ukupan broj elektrana. U sljedećem redu je ukupna potražnja za energijom. Nakon toga, u sljedećih N redova su data dva broja gdje u i -tom redu prvi broj označava moguću maksimalnu energiju elektrane a drugi broj je cijena proizvodnje po jedinici te energije. Svi brojevi su cijeli pozitivni brojevi manji od 10000000.

Izlazni podaci

Izlazna datoteka je energija.out. Sadrži samo jednu liniju sa jednim brojem, minimalnom cijenom zadovoljavanja proizvodnje.

Primjer 1

Input(energija.in)

5

21

54

73

85

45

16

Output(energija.out)

86

Primjer 2

Input(energija.in)

3

13

21

73

84

Output(energija.out)

39

Zadatak 3: Filter

Čest problem kod pravljenja programa su ulazni podaci koji su nesređeni. Tako programeri moraju prvo iz hrpe podataka izvaditi one koji su im zaista potrebni.

Vaš zadatak je da u loše formatiranom tekstu, koji sadrži mnoge nepotrebne znakove, pronađete sve cijele brojeve koji su razdvojeni i sa jedne i sa druge strane od ostatka teksta sa jednim ili više znakova razmaka. Pri tome brojevi mogu imati proizvoljan broj cifara.

Ulazni podaci

Ulazni podaci se čitaju iz tekstualne datoteke "filter.in".

U prvoj liniji datoteke se nalazi jedan cijeli broj N ($1 \leq N \leq 100.000$). U drugoj liniji se nalazi niz od N znakova koji, sa obzirom da je datoteka tekstualna, može sadržavati sva mala i velika slova engleskog alfabeta, sve cifre, razmake i specijalne znakove.

Izlazni podaci

Izlazne podatke je potrebno ispisati u datoteku "filter.out".

Potrebno je ispisati sve cijele brojeve (tj. broj ne smije sadržavati zarez ili tačku) unutar niza koji su odvojeni

od ostatka teksta znakovima razmaka. Brojeve treba ispisati u poretku u kojem se nalaze u datom nizu, svaki u posebnom redu. Ukoliko broj ima vodeće nule, ne treba ih ispisivati. Također, brojevi mogu biti samo pozitivni, tj. ako je ispred broja minus (-) broj ne treba ispisati.

Primjer 1

filter.in

36

0 -1 21 !" \$4 0.5 fs444g 333& 024 32

filter.out

0

21

24

32

Primjer 2

filter.in

26

! 031006479820153678412 d

filter.out

31006479820153678412

Ograničenja na resurse

Vaš program se treba izvršavati za ne više od 1s i ne smije koristiti više od 32 MiB memorije po svakom testnom slučaju.