Županijsko natjecanje iz informatike

Srednja škola
Druga podskupina (3. i 4. razred)

12. veljače 2016.

Zadaci

Ime zadatka	Vremensko ograničenje	Memorijsko ograničenje	Broj bodova
Popust	1 sekunda	512 MiB	30
Cache	1 sekunda	512 MiB	40
Poli	1 sekunda	512 MiB	60
Sažetko	1 sekunda	512 MiB	70
Ukupno			200

Agencija za odgoj i obrazovanje
Education and Teacher Training Agency

HRVATSKI SAVEZ INFORMATIČARA

Ministarstvo znanosti, obrazovanja i sporta

HRVATSKA ZAJEDNICA TEHNIČKE KULTURE

Zadatak: Popust

Frane kupuje mobilni telefon koji košta 1000 kuna, a na raspolaganju ima n kupona pomoću kojih može spustiti cijenu telefona. Postoje dvije vrste kupona:

1. Kupon oblika "X`kn`" smanjuje cijenu telefona za X kuna.
2. Kupon oblika "Y`%`" smanjuje cijenu telefona za Y posto.

Frane može upotrijebiti *točno tri* od njegovih n kupona i to jedan za drugim. Na primjer, ako Frane najprije upotrebi kupon "`50%`", cijena telefona se snižava na 500 kuna, ako nakon toga upotrebi kupon "`100kn`" cijena se snižava na 400 kuna, ako na kraju upotrebi kupon "`20%`" cijena se snižava na 320 kuna.

Frane je primjetio da može više uštediti ako pametno odabere ta tri kupona te redoslijed kojim ih primjenjuje. Ako, na primjer, primjeni kupon "`50%`", pa zatim "`20%`", te na kraju "`100kn`", onda će konačna cijena biti 300 kuna.

Zadani su kuponi koje Frane ima na raspolaganju. Pronađite *najmanju moguću* cijenu koju Frane treba platiti.

Ulazni podaci

U prvom redu nalazi se prirodni broj n ($3 \le n \le 10$) – broj kupona. U svakom od sljedećih n redova je zapisan jedan kupon. Svaki kupon je ili oblika "X`kn`" gdje je X prirodni broj manji od 1000 ili "Y`%`" gdje je Y prirodni broj manji od 100. Između broja X odnosno Y te znakova "`kn`" odnosno "`%`" nema razmaka.

Ulaz je takav da Frane uvijek mora platiti barem jednu kunu za telefon.

Izlazni podaci

U prvi i jedini redak izlaza ispišite jedan realni broj – najmanju moguću konačnu cijenu koju Frane treba platiti.

Bodovanje

- Rješenje za pojedini test podatak se smatra ispravnim ukoliko odstupa od službenog rješenja za najviše 0.1 kunu.
- U test podacima vrijednim 30% bodova pojavljuju se samo kuponi prvog tipa.
- U test podacima vrijednim 30% bodova pojavljuju se samo kuponi drugog tipa.

Primjeri test podataka

ulaz

```
4
50kn
50%
100kn
20%
```

izlaz

```
300.0
```

ulaz

```
3
50kn
100kn
10%
```

izlaz

```
750.0
```

ulaz

```
6
100kn
100kn
42kn
20%
10%
10%
```

izlaz

```
600.0
```

ulaz

```
4
60kn
11%
12%
14%
```

izlaz

```
673.552
```

Zadatak: Cache

Većina računala, osim radne memorije, na samom procesoru sadrži relativno malenu, ali vrlo brzu *priručnu memoriju* (takozvani *cache*) u koju procesor privremeno sprema dijelove radne memorije kojima često pristupa.

Pretpostavimo da se radna memorija računala sastoji od 2 gibibyte-a odnosno 2^{31} byte-a adresiranih brojevima od 0 do $2^{31} - 1$. Memorija je podijeljena u *blokove* veličine 64 byte-a – svaki blok počinje na adresi djeljivoj s brojem 64.

Priručna memorija se sastoji od m blokova označenih brojevima 0 do $m - 1$. Svaki blok priručne memorije također sadrži točno 64 byte-a. Na početku rada računala svi blokovi priručne memorije su prazni. Pretpostavimo da u nekom trenutku naredba želi pročitati vrijednost byte-a na adresi x, to se vrši na sljedeći način:

1. Ukoliko se u priručnoj memoriji već nalazi blok koji sadrži adresu x onda se vrijednost čita iz tog bloka priručne memorije.
2. Ukoliko se u priručnoj memoriji ne nalazi blok koji sadrži adresu x onda se cijeli blok B koji sadrži adresu x najprije čita iz radne memorije i prebacuje u priručnu memoriju, a zatim se vrijednost čita iz tog bloka priručne memorije kao u slučaju 1. Blok B se prebacuje u priručnu memoriju na sljedeći način:
   (a) Ukoliko priručna memorija nije puna, odabire se prvi po redu slobodni blok priručne memorije i u njega se kopira blok B.
   (b) Ukoliko je priručna memorija puna, iz priručne memorije se najprije *izbacuje* jedan blok te se na njegovo mjesto kopira blok B. Blok koji će se izbaciti se bira tako da se za svaki blok u priručnoj memoriji razmatra posljednja naredba koja je iz njega čitala, te se izabere onaj blok za koji je ta posljednja naredba bila najdalje u prošlosti.

Slučaj 1. se naziva *pogodak* priručne memorije, a slučaj 2. *promašaj* priručne memorije.

Zadan je niz memorijskih adresa koji se redom čitaju. Potrebno je za svaku za svaku adresu odrediti je li se dogodio pogodak ili promašaj te iz kojeg je bloka priručne memorije pročitana tražena vrijednost.

Ulazni podaci

U prvom redu nalaze se prirodni brojevi m i n ($m \leq 50, n \leq 1000$) broj blokova priručne memorije i broj adresa. U k-tom od sljedećih n redova nalazi se jedan cijeli broj x_k ($0 \leq x_k < 2^{31}$) – k-ta adresa.

Izlazni podaci

Potrebno je ispisati n redova. U k-tom redu potrebno je ispisati ispisati ili "`pogodak` b" ili "`promasaj` b", gdje je b redni broj bloka priručne memorije iz kojeg je pročitana k-ta adresa.

Bodovanje

- U test podacima vrijednim 20% bodova vrijedi $m = 2$.

Primjeri test podataka

ulaz

```
4 9
64
345
123
312
433
452
123
63
98
```

izlaz

```
promasaj 0
promasaj 1
pogodak 0
promasaj 2
promasaj 3
promasaj 1
pogodak 0
promasaj 2
pogodak 0
```

ulaz

```
2 4
11
111
22
222
```

izlaz

```
promasaj 0
promasaj 1
pogodak 0
promasaj 1
```

ulaz

```
3 6
16
55
354
783
1024
653
```

izlaz

```
promasaj 0
pogodak 0
promasaj 1
promasaj 2
promasaj 0
promasaj 1
```

ulaz

```
1 2
11
1111
```

izlaz

```
promasaj 0
promasaj 0
```

Pojašnjenje prvog primjera: Sljedeća tablica opisuje sadržaj blokova privremene memorije nakon svake pročitane adrese. Zauzeti blokovi su označeni početnom i završnom adresom bloka radne memorije koji sadrže, dok su prazni blokovi označeni s ε:

Adresa	*Rezultat*	*Blok 0*	*Blok 1*	*Blok 2*	*Blok 3*
64	promasaj 0	64 – 127	ε	ε	ε
345	promasaj 1	64 – 127	320 – 383	ε	ε
123	pogodak 0	64 – 127	320 – 383	ε	ε
312	promasaj 2	64 – 127	320 – 383	256 – 319	ε
433	promasaj 3	64 – 127	320 – 383	256 – 319	384 – 447
452	promasaj 1	64 – 127	448 – 511	256 – 319	384 – 447
123	pogodak 0	64 – 127	448 – 511	256 – 319	384 – 447
63	promasaj 2	64 – 127	448 – 511	0 – 63	384 – 447
98	pogodak 0	64 – 127	448 – 511	0 – 63	384 – 447

Zadatak: Poli

Asembler je programski jezik niže razine vrlo blizak strojnom jeziku računala. Pretpostavimo da naš procesor sadrži 10 registara označenih redom s `d0`, `d1`, ..., `d9` te razmotrimo jednostavan podskup jezika koji se sastoji od samo dvije naredbe:

`add d`a `d`b	Zbrajaju se vrijednosti registara `d`a i `d`b, rezultat se sprema u registar `d`a.
`mul d`a `d`b	Množe se vrijednosti registara `d`a i `d`b, rezultat se sprema u registar `d`a.

Neka se na početku izvođenja programa u registru `d0` nalazi broj x, a u ostalim registrima nula. U svakom trenutku vrijednost svakog registra ovisi o toj početnoj vrijednosti registra `d0` te je možemo opisati matematičkim izrazom ovisnim o varijabli x. Obzirom da u programu samo množimo i zbrajamo, taj matematički izraz mora nužno biti takozvani *polinom* i to bez slobodnog člana — odnosno izraz oblika:

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x$$

Zadan je polinom, odredite neki, *ne nužno najkraći*, program takav da je, na kraju njegovog izvođenja, vrijednost registra `d0` određena upravo zadanim polinomom.

Zadani polinom je stupnja najviše 10, a koeficijenti su pozitivni cijeli brojevi manji ili jednaki od 1000. Da bi vaše rješenje dobilo bodove za pojedini test podatak, pronađeni program smije imati *najviše 300 naredbi.*

Ulazni podaci

U prvom redu ulaza nalazi se zapis zadanog polinoma izgrađen prema sljedećim pravilima:

- Zapis polinoma se sastoji od jednog ili više *pribrojnika* odvojenih znakom "`+`". Dakle, zapis je niz znakova oblika "p_1`+`p_2`+...+`p_n".
- Svaki pojedini pribrojnik p_i je oblika "a`x^`k" gdje su *koeficijent* a i *potencija* k prirodni brojevi zapisani bez vodećih nula. Između a i k dolaze redom malo slovo '`x`' te znak potenciranja '`^`' (ASCII 94, obično `AltGr+3` na tipkovnici). Dodatno, vrijede sljedeće iznimke:
  - Ako je koeficijent a jednak 1 onda se ispušta.
  - Ako je potencija k jednaka 1 onda se ispušta i znak potenciranja "`^`" i potencija.
- Pribrojnici su poredani od većih potencija prema manjima. Niti jedna dva pribrojnika nemaju istu potenciju.
- Svaka potencija k je prirodni broj manji ili jednak od 10, a svaki koeficijent a je prirodni broj manji ili jednak od 1000.

Izlazni podaci

Ispišite redom naredbe pronađenog programa, svaku u zasebni red. Svaka naredba treba biti točno oblika "`add d`a `d`b" ili "`mul d`a `d`b", gjde su a i b znamenke. Prije oznake oba registra dolazi jedan znak razmaka.

Napomena: Rješenje ne mora biti jedinstveno.

Bodovanje

- U test podacima vrijednim 30% bodova zadani polinom je oblika "a`x`".
- U dodatnim test podacima vrijednim 30% bodova svi koeficijenti su manji ili jednaki od 10.

Primjeri test podataka

ulaz

```
5x
```

izlaz

```
add d1 d0
add d0 d1
add d0 d1
add d0 d1
add d0 d1
```

ulaz

```
8x^3+x^2
```

izlaz

```
add d1 d0
mul d1 d0
add d2 d1
mul d2 d0
add d2 d2
add d2 d2
add d2 d2
mul d0 d3
add d0 d2
add d0 d1
```

ulaz

```
2x^8+x^4+x
```

izlaz

```
add d1 d0
mul d1 d0
mul d1 d1
add d2 d1
mul d2 d1
add d0 d1
add d0 d2
add d0 d2
```

Pojašnjenje drugog primjera: Sljedeća tablica sadrži polinome koji odgovaraju vrijednostima prva četiri registra nakon svake izvedene naredbe u rješenju:

	d0	d1	d2	d3
	x	0	0	0
add d1 d0	x	x	0	0
mul d1 d0	x	x^2	0	0
add d2 d1	x	x^2	x^2	0
mul d2 d0	x	x^2	x^3	0
add d2 d2	x	x^2	$2x^3$	0
add d2 d2	x	x^2	$4x^3$	0
add d2 d2	x	x^2	$8x^3$	0
mul d0 d3	0	x^2	$8x^3$	0
add d0 d2	$8x^3$	x^2	$8x^3$	0
add d0 d1	$8x^3 + x^2$	x^2	$8x^3$	0

Zadatak: Sažetko

Frane je smislio novi način sažimanja podataka te pokušava analizirati svojstva svojeg algoritma. Osnovna ideja njegovog sažimanja je da se uklone nizovi znakova koji se više puta uzastopno ponavljaju.

Uzorak je niz od jednog ili više malih slova engleske abecede. *Složenost* uzorka je broj parova susjednih znakova u uzorku koji su različiti. Tako su, na primjer, složenosti uzoraka "`aaa`", "`abbba`" i "`baabaa`" redom 0, 2 i 3. *Poduzorak* nekog uzorka je podniz uzastonih znakova određen nekom početnom i završnom pozicijom u uzorku.

Sažetak je niz znakova koji se definira na sljedeći način:

- Svaki uzorak je ujedno i sažetak.
- Ako su s_1 i s_2 sažetci, onda je niz koji dobijemo njihovim spajanjem — "$s_1 s_2$" — također sažetak.
- Ako je s sažetak onda je i niz znakova "p`(`s`)`" također sažetak, gdje je p prirodni broj veći od jedan zapisan bez vodećih nula. Drugim riječima, od sažetka možemo dobiti novi sažetak tako da ga stavimo u zagrade te ispred otvorene zagrade napišemo prirodni broj.

Tako su, na primjer, "`ab`", "`2(ab)`" i "`2(2(ab)3(bab))`" sažetci dok "`(a)`", "`2b`" i "`abc)`" nisu.

Zadani sažetak se *proširuje* nizom koraka gdje u svakom koraku odaberemo jedan podniz sažetka oblika "p`(`s`)`", takav da niz s ne sadrži zagrade, te ga zamijenimo sa p uzastopnih primjeraka niza s. Tako se, na primjer, sažetak "`2(2(ab)3(bab))`" može najprije proširiti do "`2(abab3(bab))`" pa zatim do "`2(ababbabbabbab)`" te na kraju do `ababbabbabbababababbabbabbab`. Ako na kraju ovog postupka dobijemo uzorak u onda kažemo da je s *sažetak od* u odnosno da je u *proširenje od* s.

Zadan je sažetak s, te dva prirodna broja a i b. Neka je p proširenje od s, pronađite složenost poduzorka od p počevši od a-te pozicije pa sve do b-te pozicije (uključivo) u uzorku p.

Ulazni podaci

U prvom redu nalazi se niz od najvše 1000 malih slova engleske abecede, znamenki te znakova zagrada "`(`" i "`)`" – zadani sažetak. Za svaki prirodni broj p u strukturi sažetka vrijedi $2 \le p \le 1000$. Označimo sa n ukupnu duljinu proširenja zadanog sažetka. Za broj n vrijedi $1 \le n < 2^{31}$.

U drugom redu nalaze se prirodni brojevi a i b ($a \le b \le n$), početna i završna pozicija poduzorka.

Izlazni podaci

U prvi red ispišite jedan cijeli broj – traženu složenost.

Bodovanje

Neka n označava dužinu proširenja zadanog sažetka kao u poglavlju *Ulazni podaci*.

- U test podacima vrijednim 20% bodova vrijedi $a = 1, b = n$ i $n \le 1000$.
- U dodatnim test podacima vrijednim 20% bodova vrijedi $a = 1, b = n$.
- U dodatnim test podacima vrijednim 20% bodova vrijedi $n \le 1000$.

Primjeri test podataka

ulaz	**ulaz**	**ulaz**
`2(2(ab)3(bab))` `5 18`	`m2(i2(s))i2(p)i` `2 9`	`1000(1000(333(aba)c))` `1 1000000000`
izlaz	**izlaz**	**izlaz**
`10`	`5`	`667999999`

Pojašnjenje drugog primjera: Sažetak "`m2(i2(s))i2(p)i`" se proširuje na sljedeći način:

`m2(i2(s))i2(p)i` $\Rightarrow$ `m2(iss)ippi` $\Rightarrow$ `mississippi`

Poduzorak od 2. do 9. znaka je "`ississip`" čija je složenost jednaka 5.