

ZADATAK	BITSTOCK	ČETVEROKUT	DVAPUT	HYPERLOOP	LIJEPI PUTEVI	PARKET	PROSJEK	RAM	ŠETNJA
ulazni podaci	standardni ulaz								
izlazni podaci	standardni izlaz								
vremensko ograničenje	1 s	5 s	6 s	1 s	1 s	1 s	1 s	20 s	1 s
memorijsko ograničenje	128 MB	128 MB	128 MB	128 MB	128 MB	128 MB	128 MB	2 MB	128 MB

Mirko se ludo zabavlja novom MMORPG igrom Bitstock u kojoj igrači kupuju dionice. Na tržištu je ponuđeno točno  $N$  tipova dionica i Mirko može u bilo kojem trenutku kupiti proizvoljnu količinu bilo kojeg tipa dionica. Jedna dionica  $i$ -tog tipa košta  $c_i$  kuna i ona od trenutka kupnje nadalje generira profit to od  $g_i$  kuna **po sekundi**. Dionice je isto tako moguće kupiti u realnoj (ali pozitivnoj!) količini. Drugim riječima, ako Mirko kupi 0.7 dionice A, ona će u sljedećih 0.3 sekunde generirati  $0.21g_A$  kuna.

Posebnost Bitstocka je što neke dionice "podržavaju" neke druge koje potom možete kupiti u pola cijene. Primjerice, ako dionica A podržava B i C, tada uz posjedovanje 1 dionice A možete kupiti 0.7 dionice B i 0.3 dionice C u pola regularne cijene. No, pomoću te iste dionice A više ne biste mogli financirati daljnju kupovinu jer ste ju cijelu "iskoristili" (iako ona i dalje generira novac). Nadalje, svaki tip dionice je podržan od najviše jednog tipa dionice, te nikada **ne podržava samu sebe** (izravno ili neizravno).

Cilj igre je u što manje vremena ostvariti generiranje profita od  $P$  kuna **po sekundi**. Mirko na početku ima  $E$  kuna, ali još ne posjeduje niti jednu dionicu.

### ULAZNI PODACI

U prvome retku nalaze se prirodni brojevi  $N, E, P$  ( $1 \leq N \leq 1\,000$ ,  $1 \leq E, P \leq 1\,000\,000\,000$ ), broj dionica, početni broj kuna te tražena količina profita po sekundi.

U sljedećih  $N$  redaka nalaze cijeli brojevi  $c_i, g_i, otac_i$  ( $1 \leq c_i \leq 1\,000\,000\,000$ ,  $0 \leq g_i \leq 1\,000\,000\,000$ ,  $0 \leq otac_i \leq N$ ), cijena dionice, profit dionice po sekundi te tip dionice koja nju podržava. Ako je  $otac_i = 0$ , to označava da dionica  $i$  nije podržavana od niti jedne dionice. Barem jedan od brojeva  $g_i$  će biti pozitivan.

### IZLAZNI PODACI

Minimalno vrijeme u sekundama potrebno Mirku da ostvari ratu generiranja profita od  $P$  kuna po sekundi.

Rezultat zaokružite na cijeli broj **prema gore** (drugim riječima, ako je potrebno 17.2 sekunde, zaokružite rješenje na 18 sekundi).

### PRIMJERI TEST PODATAKA

<b>ulaz</b> 1 1 1000000 200 100 0 <b>izlaz</b> 30	<b>ulaz</b> 2 1 1000000 200 100 0 2 1 1 <b>izlaz</b> 29	<b>ulaz</b> 2 1 1000000 200 100 2 2 1 0 <b>izlaz</b> 14
---	--	--

Mirko je veliki obožavatelj tratinčica. Svaki put kada organizira piknik, on sebi izradi novi pokrivač u obliku **četverokuta**. Kada stavlja svoj pokrivač na livadu, Mirko želi da se vrhovi četverokuta nalaze točno na pozicijama tratinčica koje tamo rastu. Mirkovo je društvo veliko tako da on nastoji izraditi pokrivač **što veće površine** kako bi na njega svi stali, koristeći **najviše  $L \text{ m}^2$**  platna kojeg ima na raspolaganju.

**Napomena:** Pokrivač nije nužno konveksan četverokut, ali je uvijek pozitivne površine. Mirko je vrstan krojač pa platno može proizvoljno rezati i lijepiti kako bi dobio oblik četverokuta. Nadalje, pokrivač smije sadržavati kolinearne točke.

### ULAZNI PODACI

U prvome retku nalaze se prirodan broj  $N$  ( $4 \leq N \leq 300$ ) i cijeli broj  $L$  ( $1 \leq L \leq 10^9$ ), koji predstavljaju broj tratinčica na livadi te ukupnu površinu platna kojeg Mirko ima na raspolaganju (u  $\text{m}^2$ ).

Svaki od sljedećih  $N$  redaka sadrži dva cijela broja  $-10\,000 \leq x, y \leq 10\,000$ , koordinate tratinčica na livadi. Nijedne dvije tratinčice neće rasti jedna iz druge, tj. neće se nalaziti na istim koordinatama.

### IZLAZNI PODACI

Potrebno je ispisati maksimalnu površinu pokrivača, zaokruženu na **2 decimalna mjesta**. Ukoliko ne postoji pokrivač prema Mirkovim željama, potrebno je ispisati "**0.00**" (bez navodnika).

### PRIMJERI TEST PODATAKA

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
4 7	7 21	4 9837
-1 -1	-4 -2	-100 -100
-1 1	-1 2	-100 100
1 1	-5 -1	100 100
1 -1	7 5	100 -100
<b>izlaz</b>	2 3	<b>izlaz</b>
4.00	-10 1	0.00
	1 -2	
	<b>izlaz</b>	
	21.00	

Mirko i Slavko su čuli u zadnje vrijeme puno fantastičnih algoritama i zadataka vezanih za stringove, pa su odlučili okušati svoje znanje.

Slavko je dao Mirku jedan zadatak, u kojemu Mirko dobije string **S**, i mora reći duljinu najduljeg podstringa od **S** koji se pojavljuje u **S** na bar 2 različita mjesta.

No Slavko dobro zna da je taj zadatak prelagan za Mirka, pa mu ubrzo zadaje drugi zadatak. U drugom zadatku Slavko daje Mirku string **S** i broj **K**, te mu kaže da smije promijeniti bilo koje slovo u **S** u bilo koje drugo slovo, ali mijenjanje smije izvršiti **najviše K puta**. Sad Slavko traži od Mirka najveće rješenje prvog zadatka koje možemo dobiti mijenjanjem stringa **S**.

Mirka je zaboljela glava razmišljajući, ali nipošto ne želi priznati Slavku da postoji zadatak koji on ne zna riješiti, pa vas moli da vi riješite zadatak umjesto njega.

### ULAZNI PODACI

U prvom redu se nalazi **K** ( $0 \leq K \leq 5000$ ).

U drugom redu se nalazi string **S** koji se sastoji isključivo od malih slova engleske abecede ( $1 \leq |S| \leq 5000$ ).

### IZLAZNI PODACI

Potrebno je ispisati točno jedan broj, rješenje drugog zadatka.

### PRIMJERI TEST PODATAKA

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
1	7	2
abcba	aaaaaaaaaa	tmirkoxlyzmirkoslav
<b>izlaz</b>	<b>izlaz</b>	<b>izlaz</b>
3	9	8

**Pojašnjenje prvog test primjera:** zamjenom trećeg slova 'c' u 'a' string 'aba' pojavljuje 2 puta.

**Pojašnjenje drugog test primjera:** Mirko ne mora promijeniti niti jedno slovo.

*Napomena:* podstring od stringa **S** je bilo koji niz slova koji se uzastopno pojavljuje u **S**  
 $A = S_L S_{L+1} S_{L+2} \dots S_{R-1} S_R$ ,  $1 \leq L \leq R \leq |S|$ ,  $S_x = x$ -to slovo od stringa **S**

U kolovozu 2013. pojavio se dizajn novog načina prijevoza zvanog Hyperloop, u kojem bi kapsule putovale kroz cijev od San Francisca do Los Angelesa brzinom od tisuću kilometara na sat. Ono što, međutim, još nitko ne zna jest da će ovaj način prijevoza biti najprije ostvaren na relaciji Čazma-Daruvar. Slijedeći njihov primjer, **svaka dva** hrvatska grada, od njih ukupno **N**, bit će **izravno** povezana jednom dvosmjernom Hyperloop cijevi.

Nakon što su cijevi spremne za promet, vaš je zadatak organizirati nekoliko ruta za turistički obilazak Hrvatske. **Svaka ruta** mora **obići svih N** gradova (svaki samo jednom) i **vratiti se u početni grad** -- koristeći, dakle, ukupno **N** cijevi. Osim toga, da ne bi došlo do gužve, **dvije različite rute ne smiju koristiti istu cijev** (neovisno o smjeru).

Saznavši da je **N** neparan, izračunali ste da biste uz ove uvjete mogli organizirati mnogo ruta: toliko da nijedna cijev ne ostane neiskorištena. Preostaje vam konstruirati te rute!

### ULAZNI PODACI

U jedinome retku nalazi se neparan prirodan broj **N** ( $3 \leq N < 2000$ ), broj gradova. Gradovi su označeni brojevima od 1 do **N**.

### IZLAZNI PODACI

Ispišite  $(N - 1) / 2$  traženih ruta. Rutu ispišite kao redak od **N** razmakom odvojenih oznaka gradova koje ona posjećuje, redom od početnog grada.

**Napomena:** Rješenje ne mora nužno biti jedinstveno. Svako točno rješenje bit će prihvaćeno.

### PRIMJERI TEST PODATAKA

<b>ulaz</b>	<b>ulaz</b>
5	7
<b>izlaz</b>	<b>izlaz</b>
1 2 4 3 5	1 2 3 4 5 6 7
2 3 1 4 5	1 3 5 7 2 4 6
	1 4 7 3 6 2 5

U jednoj dalekoj zemlji postoji  $N$  gradova koji su s pomoću  $N - 1$  cesta povezani u **stablo**, što znači da između svakih dvaju gradova postoji jedinstven put. Udaljenost dvaju gradova definiramo kao broj cesta na tom putu između njih.

Neobično je što ova zemlja ima **dva glavna grada** označena brojevima **1** i **2**; ostali gradovi označeni su brojevima od 3 do  $N$ .

Mirko je dobio zadatak organizirati autobusni promet u ovoj zemlji. Da bi uspio efikasno organizirati autobuse, definirao je najprije **važnost grada** kao udaljenost toga grada od njemu bližeg glavnoga grada, a potom **važnost puta** kao najmanju važnost nekog grada na tome putu.

Vaš je zadatak zbrojiti važnosti puteva između svakih dvaju različitih gradova (takvih parova ima  $N \cdot (N - 1) / 2$ ).

### ULAZNI PODACI

U prvome retku nalazi se prirodan broj  $N$  ( $1 \leq N \leq 100\,000$ ), broj gradova. Svaki od sljedećih  $N - 1$  redaka sadrži dva međusobno različita broja  $A, B$  ( $1 \leq A, B \leq N$ ), što predstavlja cestu između gradova  $A$  i  $B$ .

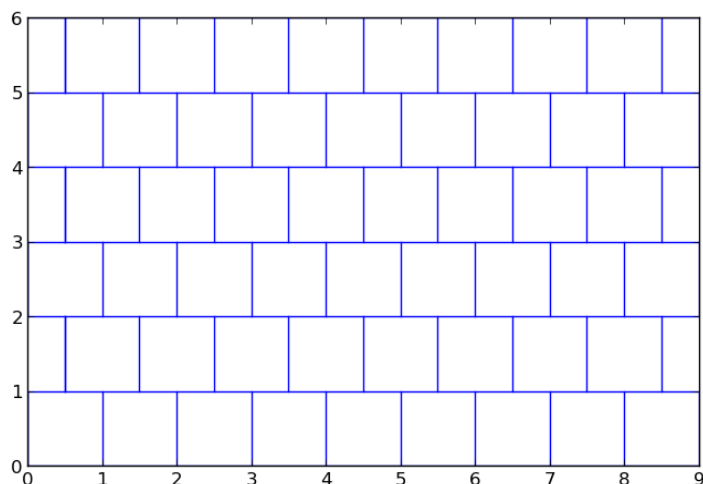
### IZLAZNI PODACI

U jedini redak ispišite traženi zbroj važnosti svih puteva.

### PRIMJERI TEST PODATAKA

ulaz	izlaz
7	7
7 2	
7 3	
7 4	
2 5	
3 6	
4 1	

Mirkov parket sastoji se od kvadratnih pločica dimenzija  $1 \times 1$  koje su složene u sljedeći uzorak:



Na slici je prikazan samo dio parketa u zamišljenom koordinatnom sustavu. Parket je zapravo toliko velik da možemo slobodno pretpostaviti da prekriva cijelu ravninu.

Za potrebe svog diplomskog rada iz matematike, Mirko je na parketu crvenom bojom nacrtao **pravokutnik**. Poslije je otkrio da boja sadrži štetnu kemikaliju koja uništava parket i sada mora promijeniti **sve pločice** koje **s nacrtanim rubom** pravokutnika imaju **barem jednu zajedničku točku**. Pomozite Mirku i recite mu: koliko ima takvih pločica?

### ULAZNI PODACI

U prvome retku nalaze se realni brojevi  $x_1, y_1$ , koordinate donjeg lijevog ruba pravokutnika na parketu.

U drugome retku nalaze se realni brojevi  $x_2, y_2$ , koordinate gornjeg desnog ruba pravokutnika. Zapis koordinata imat će najviše jedno decimalno mjesto i vrijedit će:  $0 < x_1 < x_2 \leq 10^9$ ,  $0 < y_1 < y_2 \leq 10^9$ .

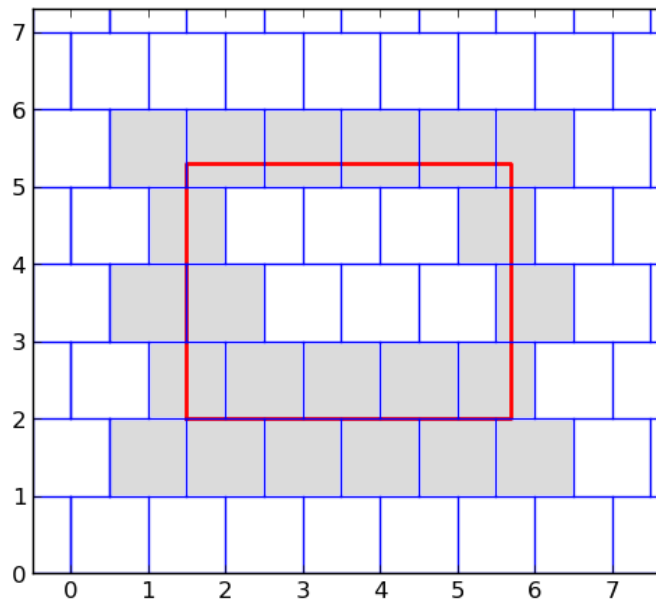
### IZLAZNI PODACI

U jedini redak ispišite broj pločica koje treba promijeniti.

**PRIMJERI TEST PODATAKA**

ulaz	izlaz
1.5 2 5.7 5.3	22

**Pojašnjenje test primjera:**





Niz brojeva najčešće zbrajamo tako da najprije zbrojimo prva dva broja, potom zbrojimo rezultat i treći broj, pa novi rezultat i četvrti broj, i tako dalje. Mirko je pomislio da na isti način može računati i **prosjeck** niza brojeva. On dakle uzme prosjek prvih dvaju brojeva, pa prosjek rezultata i trećega broja, pa prosjek novog rezultata i četvrtoga broja, i tako dalje.

Na primjer, za niz 1, 5, 9, 7 Mirko računa  $(1 + 5) / 2 = 3$ , potom  $(3 + 9) / 2 = 6$  i konačno  $(6 + 7) / 2 = 6.5$ . (Stvarni prosjek ovog niza nije 6.5, nego 5.5.)

Budući da je nedavno pogledao sve epizode Dextera i True Detectiva, Mirku nema druge nego se zabavljati s prosjecima. On ponekad promijeni dva elementa svojega niza brojeva i pokušava pogoditi je li se prosjek niza (dobiven njegovim algoritmom) smanjio, povećao ili možda ostao isti. Pomozite Mirku i napišite program koji na ova pitanja odgovara!

### ULAZNI PODACI

U prvome retku nalaze se prirodan broj  $N$  ( $2 \leq N \leq 100\,000$ ), duljina niza

U sljedećem retku nalazi se Mirkov niz:  $N$  cijelih brojeva iz intervala  $[1, 10^9]$ .

U sljedećem retku nalazi se prirodan broj  $M$  ( $1 \leq M \leq 100\,000$ ), broj promjena niza.

Svaki od sljedećih  $M$  redaka ima oblik " $p A q B$ " i opisuje promjenu: postavi  $p$ -ti element niza na  $A$ , a  $q$ -ti element na  $B$ . Pritom je  $1 \leq p, q \leq N$ ,  $p \neq q$  i  $1 \leq A, B \leq 10^9$ . Promjene su kumulativne, tj. niz ostaje promijenjen.

### IZLAZNI PODACI

Za svaku promjenu u zaseban redak ispišite znak "<" ako se prosjek smanjio, ">" ako se povećao, te "=" ako je ostao isti.

### PRIMJERI TEST PODATAKA

ulaz	izlaz
6	=
20 40 50 60 100 25	<
4	=
1 55 2 5	>
1 135 6 10	
6 20 4 20	
3 100 4 83	

#### Pojašnjenje:

20, 40, 50, 60, 100, 25 -> 50.0

55, 5, 50, 60, 100, 25 -> 50.0

135, 5, 50, 60, 100, 10 -> 45.0

135, 5, 50, 20, 100, 20 -> 45.0

135, 5, 100, 83, 100, 20 -> 56.0

Mirku su hakeri provalili u računalu iskorištavanjem Shellshock propusta te podizanjem voltaže u sustavu uništili gotovo sav RAM osim posljednja 2MB. Mirko u računalu ima točno 26 tvrdih diskova označenih velikim slovima engleske abecede od 'A' do 'Z'. Na sreću, Mirko ima **ogroman** dnevnik pristupa podacima na tvrdim diskovima koji se sastoji od niza oznaka tvrdih diskova. Analizirao je napad hakera na sljedeći način:

- Dnevnik je podijelio na **N** manjih datoteka **S<sub>1</sub>** do **S<sub>N</sub>**, koje može učitati u RAM. Svaka od njih je niz velikih slova engleske abecede.
- Datoteke je učitavo jednu za drugom te je nakon učitane datoteke **S<sub>i</sub>** izračunao **približno** koliko puta su hakeri pristupili tvrdom disku **H<sub>i</sub>** u posljednjih **K<sub>i</sub>** pristupa u cijelom dnevniku do datoteke **S<sub>i</sub>**, uključivo.

Mirko je ponosan jer je vješto uspio izračunati odgovore na sve upite koji su ga zanimali te razotkrio namjere hakera. Kako vas hakeri ne bi uhvatili nespremne, vaš izazov je napisati program koji će dati **približne** odgovore na Mirkove upite uz samo 2MB RAM-a.

### ULAZNI PODACI

U prvom retku nalazi se prirodan broj **N** ( $1 \leq N \leq 200\,000$ ).

Sljedećih **2N** redaka podijeljeni su na **N** grupa od 2 retka:

- U prvome od njih nalazi se niz velikih slova engleske abecede **S<sub>i</sub>** ( $1 \leq |S_i| \leq 4\,000$ )
- U drugome se nalaze znak **H<sub>i</sub>** (veliko slovo engleske abecede), oznaka tvrdog diska te broj pristupa **K<sub>i</sub>** ( $1 \leq K_i \leq \sum_{j=1}^i |S_j|$ )

**Napomena:** Ukupna količina podataka u ulazu bit će manja od 200MB.

### IZLAZNI PODACI

U **N** redaka izlaza ispišite **približne** odgovore na **N** Mirkovih upita. Neka je **A<sub>i</sub>** vaš odgovor, a **T<sub>i</sub>** točan odgovor. Vaš odgovor smatramo **približnim** ako vrijedi  $T_i \leq A_i < 2T_i$ . Iznimno, ako je točan odgovor **T<sub>i</sub>** jednak nuli, tada i vaš odgovor **A<sub>i</sub>** mora biti jednak nuli.

**Napomena:** Memorijsko ograničenje za ovaj zadatak je samo 2MB.

### PRIMJERI TEST PODATAKA

ulaz	izlaz
3	1
BAAB	3
B 2	0
AABB	
A 6	
ZA	
Z 1	

**Pojašnjenje:** Potpuni dnevnik možemo dobiti spajanjem datoteka:

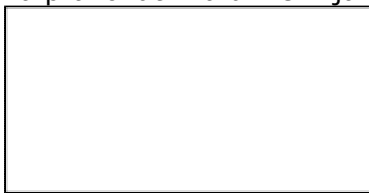
BAAB|AABB|ZA

Odgovor na 1. upit je broj pristupa tvrdom disku B u označenom podnizu: BA**AB**|AABB|ZA

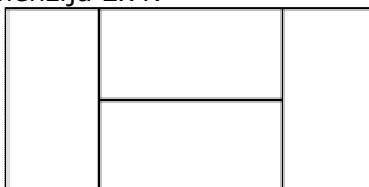
Odgovor na 2. upit je broj pristupa tvrdom disku A u označenom podnizu: BA**AB**|**AABB**|ZA

Odgovor na 3. upit je broj pristupa tvrdom disku Z u označenom podnizu: BAAB|AABB|**ZA**

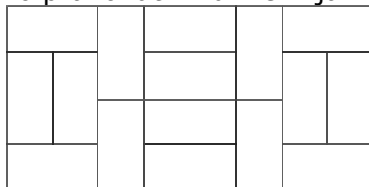
Slavko je upravo popločao svoju kuhinju na matematički zanimljiv način. Na početku je njegova kuhinja jedna pločica oblika pravokutnika dimenzija  $1 \times 2$  kao na slici:



Zatim je  $N$  puta podijelio svaku pločicu na 4 manje pločice na sljedeći način. Nakon jedne podjele dobivamo pravokutnik dimenzija  $2 \times 4$ :



Nakon druge podjele tako dobivamo pravokutnik dimenzija  $4 \times 8$ :



Nakon konačnog popločenja kuhinju možemo promatrati kao koordinatni sustav gdje svaka pločica pokriva točno **dva polja**. Polje u gornjem lijevom kutu nalazi se u prvom retku i prvom stupcu te ima koordinate  $(1, 1)$ , dok polje u donjem desnom kutu ima koordinate  $(2^N, 2^{N+1})$ .

Nakon što je popločio kuhinju Slavko je prošetao po njenim **poljima** od početnog polja  $(R, S)$  nizom pomaka na jedno od 4 polja susjedna trenutačnome. Pomake predstavljamo znakovima:

- 'L' za pomak na susjedno polje lijevo.
- 'R' za pomak na susjedno polje desno.
- 'U' za pomak na susjedno polje gore.
- 'D' za pomak na susjedno polje dolje.

Nakon svakog Slavkovog pomaka odredite je li prešao između dvije pločice.

### ULAZNI PODACI

U prvom retku nalazi se prirodan broj  $N$  ( $0 \leq N \leq 20$ ).

U drugom retku ulaza nalaze se dva prirodna broja  $R$  i  $S$  ( $1 \leq R \leq 2^N$ ,  $1 \leq S \leq 2^{N+1}$ ), redak i stupac polja u kojemu se Slavko na početku nalazi.

U trećem retku ulaza nalazi se niz Slavkovih pomaka, označenih znakovima 'L', 'R', 'D' i 'U'. Niz pomaka neće biti duži od 100 000 znakova. Slavko neće napraviti pomak izvan pravokutnika.

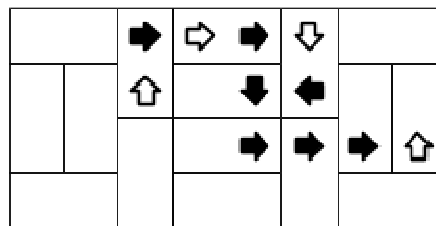
### IZLAZNI PODACI

U jedinom retku ispišite niz znakova tako da je  $i$ -ti znak jednak 'Y' ako je Slavko  $i$ -tim pomakom prešao između dvije pločice, a 'N' ako je ostao na istoj pločici.

**PRIMJERI TEST PODATAKA**

ulaz	izlaz
2	NYNYNYYYYYN
2 3	
URRRDLDRRRU	

**Pojašnjenje prvog test primjera:** Slavkov put izgleda ovako:



Zacrtnjenim pomacima Slavko je prešao između dvije pločice.