

Mirko i Slavko bave se izradom mobilnih aplikacija. Nakon izrade neočekivano dobro prihvaćene aplikacije FotoLektira koja na temelju nekoliko rečenica radi stilsku analizu knjige, ekipa je odlučna napraviti još jednu uspješnicu.

Prototip nove igre koju razvijaju ima **R** x **S** ploču na kojoj su polja ili prazna ili zablokirana. Jedno od polja sadrži jabuku koja uvijek pada u smjeru gravitacije. U svakoj sekundi igrač može ekran okrenuti za 90 stupnjeva u smjeru kazaljke na satu, za 90 stupnjeva obrnuto od smjera kazaljke na satu ili može pričekati (ne napraviti ništa). Nakon toga jabuka padne jedno polje prema dolje (u smjeru gravitacije), ali samo ako polje ispod jabuke postoji (nije izvan ploče) i prazno je - inače jabuka ostaje na mjestu.

Vaš zadatak je napraviti simulator igre, odnosno za početno stanje i niz koraka ispisati konačno stanje. Sljedeća tablica ilustrira jedan jednostavni primjer ove igre. U tablici znak '#' označava zablokirano polje, znak '.' označava prazno polje, dok znak 'J' označava poziciju jabuke.

Početno stanje	Akcija: Okret u smjeru kazaljke na satu	Akcija: čekanje	Akcija: čekanje (jabuka ostaje na mjestu)
#.. J.. #.. ..##.# ...J. .#...	..#.##.J.	..#.##.J.
Akcija: okret obrnuto od smjera kazaljke na satu	Akcija: čekanje (jabuka ostaje na mjestu)	Akcija: okret obrnuto od smjera kazaljke na satu	Konačno rješenje
#.. ... #.J ..# ...	#.. ... #.J ..##. ..J.. #.#..	...#. ..J.. #.#..

ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi **R** i **S** ($3 \leq R, S \leq 1000$), broj redaka i stupaca ploče. U sljedećih **R** redaka nalazi se niz od točno **S** znakova gdje je svaki znak veliko slovo 'J', '.' (točka) ili '#'. Na ploči će se nalaziti **točno jedan** znak 'J'.

U posljednjem retku ulaza nalazi se niz od najviše 1 000 000 znakova -- niz koraka. Svaki znak toga niza bit će jedan od sljedećih:

- 'R' - označava okret ekrana u smjeru kazaljke na satu
- 'L' - označava okret ekrana u smjeru obrnutom od smjera kazaljke na satu
- 'P' - označava čekanje

IZLAZNI PODACI

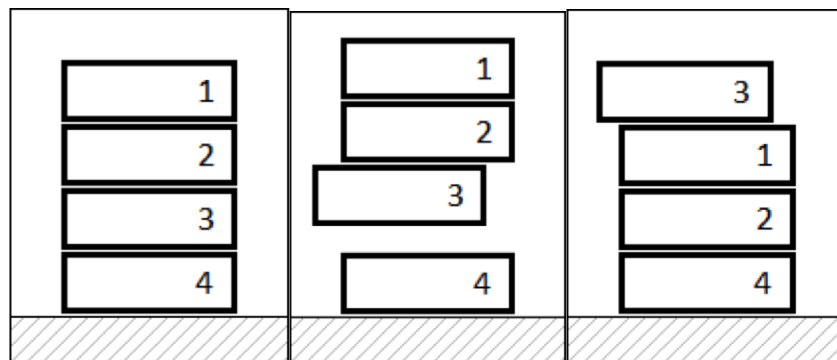
Ispišite **R** x **S** ili **S** x **R** tablicu (ovisno o konačnoj rotaciji) krajnjeg stanja ploče.

PRIMJERI TEST PODATAKA

ulaz	izlaz
5 3	...#.
#..	..J..
J..	#.#..
#..	
..#	
...	
RPPLPL	

Unatoč tome što je uzoran učenik, Mirko je uvijek imao problema s čitanjem lektira. Ove godine odlučio je tome stati na kraj. Nabavio je svih N knjiga koje profesorica može odabrati za lektire, označene prirodnim brojevima od 1 do N . Zatim ih je sve smjestio u jedan toranj od najmanje do najveće, tj. knjiga s oznakom 1 je na vrhu tornja.

Svaki put kada mu profesorica zada za lekturu knjigu L_i , Mirko je mora izvući iz tornja kako bi je pročitao, što obavlja tako da podigne sve knjige koje se nalaze iznad knjige L_i , izvuče knjigu L_i , te spusti podignute knjige na toranj. Nakon što Mirko pročita knjigu L_i , vraća je na vrh tornja. Knjigu L_i također smatramo podignutom.



Mirko čita lekturu s oznakom 3

Na primjer, pogledajmo toranj na slici. Ako je prva lektura knjiga označena brojem 3, Mirko će podići tri knjige označene brojevima 1, 2 i 3. Zatim će spustiti knjige označene brojevima 1 i 2. Nakon što pročita knjigu označenu brojem 3, stavit će je na vrh tornja.

Poznat je niz lektira L koje će profesorica zadati. Izračunajte koliko će knjiga Mirko podići ove školske godine kako bi pročitao sve lektire.

ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi N i Q ($1 \leq N, Q \leq 100\,000$), broj knjiga i broj lektira. U drugom retku nalazi se niz od Q prirodnih brojeva L ($1 \leq L_i \leq N$), zadane lektire.

Napomena: Moguće je da neka knjiga nikad ne bude odabrana za lekturu. Ujedno je moguće da neka knjiga bude odabrana više puta, u kojem slučaju je Mirko mora ponovo pročitati.

IZLAZNI PODACI

U prvom retku ispišite točno jedan prirodan broj, ukupan broj knjiga koje će Mirko podići.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
4 3	2 8	10 3
1 3 2	2 1 2 1 2 2 2 1	10 10 10
izlaz	izlaz	izlaz
7	14	12

Pojašnjenje prvog test primjera:

(1,2,3,4) -> (1,2,3,4), podignuta je samo knjiga 1

(1,2,3,4) -> (3,1,2,4), podignute su knjige 1, 2 i 3

(3,1,2,4) -> (2,3,1,4), podignute su knjige 1, 2 i 3

Ukupno je Mirko podigao $1+3+3=7$ knjiga.

Mirko je zamislio supertajni prirodan broj X koji u sebi skriva tajnu upravljanja svemirom. Njegov najbolji prijatelj Slavko opijen je moći i želi se domoći toga broja. Tijekom kasnonoćnog prekopavanja Mirkovih ladica, naišao je na papirić na kojem je objašnjeno da je suma broja X i njemu zrcalnog broja točno Y . Zrcalni broj od X tvorimo tako da unatrag zapišemo njegove dekadske znamenke. Tako je primjerice zrcalni broj od 123 jednak 321, dok je zrcalni broj od 520 jednak 25.

Za zadani broj Y izračunajte koliko prirodnih brojeva X zadovoljava gornju konstrukciju.

ULAZNI PODACI

U prvom retku nalazi se T ($1 \leq T \leq 500$), broj testnih primjera na koje morate odgovoriti. U svakom od sljedećih T redaka nalazi se po jedan prirodan broj Y ($1 \leq Y \leq 10^{18}$), opisan u tekstu zadatka.

IZLAZNI PODACI

Za svaki od T brojeva Y potrebno je ispisati jedan redak: koliko različitih brojeva X generira upravo taj broj Y .

Napomena: rješenje za pojedini Y potpuno je neovisno o prethodnim ili sljedećim redcima ulaznih podataka.

PRIMJERI TEST PODATAKA

ulaz	izlaz
4	1
10	1
11	9
121	0
109	

Pojašnjenje test primjera:

$$10 = 5 + 5$$

$$11 = 10 + 1$$

$$121 = 110 + 11 = 92 + 29 = 83 + 38 = 74 + 47 = 65 + 56 = 56 + 65 = 47 + 74 = 38 + 83 = 29 + 92$$

Mirko je na sajmu kupio novi pametni Roomba usisivač. Kako bi ga isprobao, napravio je kutiju od kartona i njeno dno podijelio na $(N + 1)$ redaka i stupaca označenih brojevima od 0 do N . Na svako polje u kutiji Mirko je postavio određeni broj novčića. U kutove, odnosno na polja $(0, 0)$, $(0, N)$, $(N, 0)$ i (N, N) stavio je zlatne novčiće, a u preostala polja srebrne. Usisivač se na početku nalazi na polju $(0, 0)$ te se u jednoj sekundi može pomaknuti na jedno od (najviše) osam susjednih polja.

Mirko je zapovjedio svome pametnom usisivaču da pokupi **sve** zlatne novčiće i što više srebrnih novčića te da se za točno $4 \cdot N$ sekundi vrati na polje s kojeg je i krenuo. Međutim, usisivač je samo tužno odgovorio da ne zna izračunati putanju kojom će pokupiti najviše novčića.

Napišite program koji će izračunati najveći broj novčića koje usisivač može pokupiti.

ULAZNI PODACI

U prvome retku nalazi se prirodan broj N ($1 \leq N \leq 500$).

U sljedećih $N + 1$ redaka nalazi se po $N + 1$ prirodnih brojeva, koji predstavljaju broj novčića na pojedinom polju kutije. Broj novčića u svakom polju bit će manji ili jednak 10000.

IZLAZNI PODACI

U prvi i jedini redak treba ispisati najveći broj novčića koje usisivač može skupiti u $4 \cdot N$ sekundi tako da se na kraju vrati na početnu poziciju i pokupi sve zlatne novčiće.

PRIMJERI TEST PODATAKA

ulaz	ulaz
2	3
1 1 1	1 2 3 4
1 10 1	2 3 4 5
1 1 1	3 4 5 6
	4 5 6 7
izlaz	izlaz
17	51

Kad je Hrvatski savez informatičara objavio rang listu s prvoga kruga ovog natjecanja, jedan mladi gospodin prišuljao se i pitao: "A gdje je pečat?"

Da se to ne bi opet dogodilo, ovaj put na rang listu utisnut ćemo pečat ne jednom, nego **dvaput**. Pečat ima oblik **konveksnoga poligona** i mjesta na kojima je udaren mogu se preklapati. Vaš je zadatak izračunati **ukupni opseg** dijela papira prekrivenog otiskom pečata.

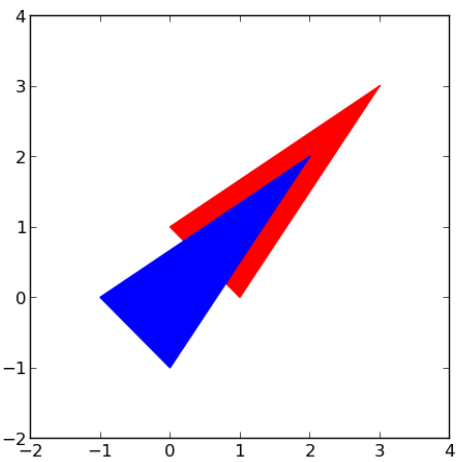
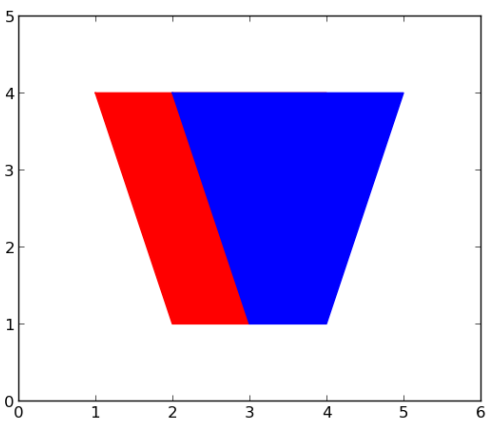
ULAZNI PODACI

U prvome retku nalazi se prirodan broj N ($3 \leq N \leq 100\,000$), broj vrhova pečata. Svaki od sljedećih N redaka sadrži cijele brojeve x, y ($0 \leq x, y < 10^9$), koordinate vrha prvoga pečata utisnutog na papir. Vrhovi su dani u smjeru kazaljke na satu i nikoja tri nisu kolinearna. U sljedećem retku nalaze se cijeli brojevi Dx, Dy ($-10^9 < Dx, Dy < 10^9$) koji čine vektor za koji je drugi pečat pomaknut u odnosu na prvi.

IZLAZNI PODACI

U jedini redak ispišite traženi opseg. Dozvoljeno je odstupanje od službenog rješenja za 0.001.

PRIMJERI TEST PODATAKA

ulaz	izlaz	pojašnjenje
<pre>3 3 3 1 0 0 1 -1 -1</pre>	12.07544	
<pre>4 1 4 4 4 3 1 2 1 1 0</pre>	12.32456	

Informatičarske zvijezde u usponu, Mirko i Slavko, često krata vrijeme smišljajući nove algoritme. Trenutno se zabavljaju sortiranjem. Mirko je predložio sljedeći algoritam:

```
while (!sorted(A)) {
    int i = random(N);
    int j = random(N);
    if (A[min(i,j)] > A[max(i,j)])
        swap(A[i], A[j]);
}
```

Slavko, inspiriran Mirkovom verzijom, predložio je:

```
while (!sorted(A)) {
    int i = random(N-1);
    int j = i + 1;
    if (A[i] > A[j])
        swap(A[i], A[j]);
}
```

Funkcija *random(k)* vraća cijeli broj iz intervala $0, 1, \dots, k-1$ gdje svaki broj ima jednaku vjerojatnost pojavljivanja.

Sada ih zanima koji je algoritam bolji. Za zadani niz **A** duljine **N** odredite očekivani broj koraka koji će biti potreban pojedinom algoritmu da se izvrši. Jednim korakom podrazumijevamo jednu potpunu iteraciju *while* petlje.

ULAZNI PODACI

U prvom retku nalazi se prirodan broj **N** ($1 \leq N \leq 8$), broj elemenata niza **A**. U sljedećem retku nalazi se **N** cijelih brojeva **A**₁, **A**₂, ..., **A**_n ($0 \leq A_i \leq 100$), elementi niza **A**.

IZLAZNI PODACI

U prvi redak ispišite očekivani broj koraka algoritma koji je Mirko predložio, a u drugi ispišite očekivani broj koraka algoritma koji je Slavko predložio. Brojeve ispišite sa šest znamenaka iza decimalne točke.

Napomena: Dozvoljeno je odstupanje od službenog rješenja za najviše 10^{-6} .

PRIMJERI TEST PODATAKA

ulaz 3 3 2 1 izlaz 6.000000 5.000000	ulaz 6 2 7 1 8 2 8 izlaz 33.000000 14.375000
---	---

Mirko stoji na ulazu, a Slavko na izlazu iz tunela. Oni zapisuju registracijske oznake automobila redom kojim oni prolaze pokraj njih i te podatke javljaju policijskoj patroli koja se nalazi nekoliko kilometara dalje.

Na osnovi Mirkovih i Slavkovih podataka policijska patrola može nepogrešivo utvrditi da su neki vozači pretjecali u tunelu, što je naravno strogo zabranjeno.

Napišite program koji će odrediti broj vozača za koje se sa sigurnošću može tvrditi da su počinili navedeni prekršaj.

Pretpostavljamo da je promet u tunelu protekao normalno, tj. bez sudara, bez zaustavljanja itd.

ULAZNI PODACI

U prvome retku nalazi se prirodan broj N ($1 \leq N \leq 1000$), broj automobila.

U sljedećih N redaka nalaze se registracijske oznake tih automobila, redosljedom kojim su ulazili u tunel.

U sljedećih N redaka nalaze se registracijske oznake tih automobila, redosljedom kojim su izlazili iz tunela.

Registracijska oznaka nekog automobila sastoji se od najmanje šest i najviše osam znakova, a jedini dozvoljeni znakovi su velika slova engleske abecede (A-Z) i znamenke dekadskog sustava (0-9).

IZLAZNI PODACI

U prvi i jedini redak treba ispisati broj vozača koje policija sigurno može kazniti zbog nepropisnog pretjecanja.

PRIMJERI TEST PODATAKA

ulaz	ulaz
3	5
ZG100LK	ZG508OK
OS1234K	PU305A
ST999D	RI604B
ZG100LK	ZG206A
ST999D	ZG232ZF
OS1234K	PU305A
	ZG232ZF
izlaz	ZG206A
1	ZG508OK
	RI604B
	izlaz
	3

Mirko i Slavko stalno igraju neke igre. Budući da su im sve postojeće dosadile, odlučili su izmisliti svoju vlastitu igru. Mirko zamisli i zapiše veliku tablicu jednoznamenkastih brojeva, a Slavko mora brzo odgovarati na pitanja oblika: *Kolika je suma brojeva u nekom pravokutniku?* Kako Mirko ne bi trebao zapisivati sve brojeve iz zamišljene tablice, on se sa Slavkom dogovorio da će svaki redak tablice biti konstruiran tako da mnogo puta slijepi osnovni period niza i skrati ga odbacujući elemente s kraja da bude širok kao i tablica.

ULAZNI PODACI

U prvom retku nalaze se brojevi **R** i **S** ($1 \leq R, S \leq 100\,000$), broj redaka i broj stupaca tablice. U sljedećih **R** redaka nalaze se osnovni periodi svakog od redaka. Svaki period sastoji se od najviše $\min(S, 100)$ znamenaka 0-9, zapisanih jedna za drugom bez razmaka. Sljedeći redak sadrži broj **Q** ($1 \leq Q \leq 100\,000$), broj pitanja na koje Slavko mora odgovoriti. Svaki od sljedećih redaka sadrži četiri cijela broja **r₁**, **s₁**, **r₂**, **s₂** ($1 \leq r_1 \leq r_2 \leq R$ i $1 \leq s_1 \leq s_2 \leq S$), opise pravokutnika u kojima Slavko mora pronaći sumu brojeva. Pritom **r₁** i **s₁** opisuju gornje lijevo polje (redak **r₁** i stupac **s₁**) u pravokutniku, dok **r₂** i **s₂** opisuju donje desno polje. Redci tablice označeni su brojevima od 1 do **R** odozgo prema dolje, a stupci brojevima od 1 do **S** s lijeva na desno.

IZLAZNI PODACI

U **Q** redaka moraju se nalaziti Slavkovi odgovori, u poretku u kojem ih je davao, u svakom retku točno jedan broj.

PRIMJERI TEST PODATAKA

ulaz	ulaz
4 10	4 3
1	5
04	0
123	330
98	405
1	2
1 1 4 10	1 1 4 1
izlaz	1 1 3 1
134	izlaz
	12
	8

Pojašnjenje prvog test primjera: tablica opisana periodima danim na ulazu izgleda ovako:

```
1111111111
0404040404
1231231231
9898989898
```

Slavko mora odgovoriti na jedno jedino pitanje koje traži sumu svih brojeva u toj tablici što iznosi 134.

Nedavno je u Hrvatskoj nastao novi grad zvan Xor. Grad se sastoji od N raskrižja te M dvosmjernih cesta. Svaka cesta povezuje par raskrižja te ima duljinu, nenegativni cijeli broj. Mirko se nedavno preselio u Xor grad te planira često koristiti taksije. Za Xor grad svojstveno je da se cijena taksi vožnje računa na poseban način: koristi se **bitovna XOR operacija** nad duljinama cesta. Na primjer, ako vožnja prolazi cestama duljina 3, 4 i 6, cijena je $3 \text{ XOR } 4 \text{ XOR } 6 = 1$. Tijekom vožnje taksi smije proći istom cestom više puta, a **svaki prolaz računa se u cijenu**.

Mirko želi isplanirati niz vožnji taksijem tako da potroši što manje na sve vožnje. Svaka vožnja zadana je polaznim raskrižjem A i dolaznim raskrižjem B , a potrebno je odrediti **najmanju moguću cijenu** vožnje. Taksi smije doći u raskrižje B i nastaviti vožnju, no mora naknadno završiti vožnju u B (vidi prvi primjer).

ULAZNI PODACI

U prvom retku nalaze se brojevi N i M ($1 \leq N, M \leq 200\,000$), broj raskrižja i broj cesta. U sljedećih M redaka se nalaze opisi cesta. U svakom retku nalaze se tri broja A , B i C ($1 \leq A < B \leq N$; $0 \leq C \leq 1\,000\,000\,000$) koji označuju da cesta povezuje raskrižja A i B , a duljina joj je C .

Nikoje dvije ceste neće povezivati isti par raskrižja.

Iz svakog raskrižja bit će moguće doći do svakog drugog raskrižja.

U sljedećem retku nalazi se prirodan broj Q ($1 \leq Q \leq 200\,000$), broj Mirkovih vožnji.

U sljedećih Q redaka nalaze se vožnje. Svaka vožnja opisana je parom brojeva A i B ($1 \leq A, B \leq N$), gdje je A polazno raskrižje te B dolazno.

IZLAZNI PODACI

Za svaku od Q zadanih vožnji, ispišite u jedan redak najmanju moguću cijenu vožnje. Cijene je potrebno ispisati u poretku u kojem su vožnje bile zadane na ulazu.

Napomena: $A \text{ XOR } B$ na i -toj poziciji u bazi 2 ima jedinicu ako i samo ako su znamenke na i -toj poziciji u A i B različite (npr. $0011 \text{ XOR } 0101 = 0110$).

PRIMJERI TEST PODATAKA

ulaz	ulaz
5 5	4 6
1 2 3	1 2 7
2 3 9	1 3 2
3 4 2	1 4 0
4 5 6	2 3 13
3 5 7	2 4 19
1	3 4 31
1 2	3
izlaz	4 1
0	4 2
	4 3
	izlaz
	0
	6
	2

Pojašnjenje prvog test primjera: raskrižja na kojima će se Mirkov taksi pojaviti: (1, 2, 3, 4, 5, 3, 2), cijena = $3 \text{ XOR } 9 \text{ XOR } 2 \text{ XOR } 6 \text{ XOR } 7 \text{ XOR } 9 = 0$