

ZADATAK	BOMBONI	CESTE	ČESTICE	IGRA	PREKIDAČ	RIVA	RSA	TROKUT	ZMIJA
ulazni podaci	standardni ulaz								
izlazni podaci	standardni izlaz								
vremensko ograničenje	2 s	1 s	5 s	1 s	10 s	2 s	2 s	1 s	4 s
memorijsko ograničenje	64 MB								

Ivica je pozvao sve svoje prijatelje na veliku zabavu. Svi znaju da nijedno pravo druženje ne može proći bez megapopularnih i nezamjenjivih PEZ bombona. Ivica se boji da će tih slatkiša biti premalo, pa ne želi kupiti manje od A bombona. Isto tako, ima dovoljno novaca za najviše B bombona.

Budući da se neće svi prijatelji odazvati, Ivica ne zna točan broj sudionika zabave. Stoga želi kupiti količinu bombona koju može ravnopravno podijeliti na najviše moguće načina. Podjela bombona je ravnopravna ako svaki od prijatelja dobije jednako mnogo bombona i ako pri tome svi kupljeni bomboni budu podijeljeni. Na primjer, ukoliko Ivica kupi 6 bombona, moći će ih ravnopravno podijeliti na 4 načina:  $\{ 1+1+1+1+1+1, 2+2+2, 3+3, 6 \}$ .

Napišite program koji će pomoći Ivici odlučiti koliko bombona treba kupiti.

### Ulazni podaci

U prvom i jedinom retku ulaza nalaze se dva prirodna broja A i B,  $1 \leq A \leq B \leq 2\,000\,000$ , odvojena jednim razmakom.

### Izlazni podaci

Označimo sa M maksimalni broj načina na koje Ivica može ravnopravno podijeliti kupljene bombone. Neka je S skup svih brojeva X u intervalu  $[A, B]$  takvih da ako Ivica kupi X bombona, onda ih može ravnopravno podijeliti na M načina; neka je N broj elemenata skupa S.

U prvi redak izlaza ispišite brojeve M i N odvojene razmakom. U svaki od idućih N redaka ispišite po jedan element skupa X; elementi trebaju biti ispisani uzlazno po veličini.

### Primjeri test podataka

<b>ulaz</b> 1 10	<b>ulaz</b> 13 15	<b>ulaz</b> 9125 1912412
<b>izlaz</b> 4 3 6 8 10	<b>izlaz</b> 4 2 14 15	<b>izlaz</b> 288 4 1441440 1663200 1801800 1884960

Cestovni sustav nekog grada, gledan iz zraka, izgleda kao polje od  $R \times S$  jediničnih kvadratića. Svaki jedinični kvadratić može biti ili prazan ili se u njemu može nalaziti jedinični dio neke ceste. Jedinični dio ceste može biti vertikalna, dijagonalna, horizontalna ili pak raskrižje. Na donjoj slici je prikazan jedan takav cestovni sustav:

```

.....
.....+---+---+---+---+..
...../.../... \... |..
.....+---+---+---+---+..
.....| \./ | .....+---+..
.....|.+.|. .....|. .....
.....|/. \ | .....+. .....
.....+---+---+---+---+..
..... \... / .....
.....+---+---+---+---+..

```

U polju, dakle, postoji 6 dozvoljenih znakova: znak ' . ' (točka) predstavlja prazni jedinični kvadratić kroz kojeg ne prolazi cesta. Znak ' - ' (minus) predstavlja horizontalnu jediničnu cestu, znakovi ' / ' i ' \ ' predstavljaju dijagonalne jedinične ceste, dok znak ' | ' („pipe“) predstavlja vertikalnu jediničnu cestu. Znak ' + ' koji predstavlja raskrižje.

U sljedećoj tablici je precizno opisano sa kojih od susjednih 8 jediničnih kvadratića automobil može ući na pojedinu jediničnu cestu, odnosno izaći sa nje:

## # # ##	### *- ###	##* #/# *##	*## #\# ##*	*** *+* ***
-----------------	------------------	-------------------	-------------------	-------------------

Znakom ' # ' označili smo jedinične kvadratiće sa kojih se ne može ući na središnju jediničnu cestu. To su ujedno i kvadratići na koje se ne može izaći iz središnje jedinične ceste. Znakom ' \* ' označili smo one jedinične kvadratiće sa kojih je to moguće.

**Siguran cestovni sustav** mora poštovati određena pravila: u njemu ne smije postojati slijepa ulica, odnosno jedinična cesta koja je spojena na manje od dvije susjedne jedinične ceste. Za dvije jedinične ceste kažemo da su spojene ukoliko se može doći (gledajući gornju tablicu) sa prve na drugu te sa druge na prvu. Cestovni sustav na gornjoj slici je siguran.

Naš prijatelj Ljubo je opet počeo raditi nepodopštine: ove godine je bespravno sagradio nekoliko cesta. Zbog takve gradnje, cestovni sustav možda više nije siguran. Vaš je zadatak učiniti sustav ponovno sigurnim za promet uništavajući što je moguće manje jediničnih cesta. Uništenje jedne jedinične ceste podrazumijeva pretvaranje odgovarajućeg znaka u ' . ' (točku).

### Ulazni podaci

U prvom redu nalaze se brojevi  $R$  i  $S$ ,  $1 \leq R \leq 20$ ,  $1 \leq S \leq 20$ , broj redaka i stupaca cestovnog sustava. U svakom od sljedećih  $R$  redaka nalazi po  $S$  znakova koji predstavljaju sustav na način opisan u tekstu. Pri tome će se javljati samo 6 gorenavedenih znakova.

### Izlazni podaci

U R redaka treba ispisati S znakova, koji predstavljaju popravljenu cestovni sustav tako da on bude siguran. Na originalnom cestovnom sustavu ne smijete raditi nikakve promjene osim pretvaranja neke jedinične ceste u znak ' . ' (točku).

**Napomena:** Može se pokazati da će siguran cestovni sustav sa najmanjim mogućim brojem uništenih cesta biti jedinstven.

### Primjeri test podataka

```
ulaz
5 6
...../.
.+ + + ..
.| . | ..
.+ + + .
... + ..

izlaz
.....
.+ + + ..
.| . | ..
.+ + + ..
.....
```

```
ulaz
10 10
.....+.
..+.+.
..+.+.
.....+.
.....
++-+---+.
||.....|.
+\...../.
..\--/.....
...--.....

izlaz
.....
..+.
..+.
.....
.....
.....
.....
.....
.....
.....
```

```
ulaz
10 20
.|.....
.+...+---+---+---+.
.|.../.../...\.|..
.|.+---+.....\.|..
.|.\./|.....+---+.
.|.|.+|. + + + .|.|.
.|.|./.\|.....+.|.
.|.+---+...../.\|.
.|.....\.../...\.|..
.|...+ + + ..+ + + ..

izlaz
.....
.....+---+---+---+.
...../.../...\.|..
.....+---+.....\.|..
.....|\./|.....+---+.
.....|. + .|.|.|.
.....|./.\|.....+.
.....+---+...../.....
.....\.../.....
.....+ + + ..
```

Mirko se preko veze zaposlio kao fizičar u Europskoj Organizaciji za Nuklearna Istraživanja (CERN) te je kao prvi zadatak dobio izradu nacrtu za najnoviji ubrzivač čestica. Ubrzivač će imati točno **N komora** u kojima se na početku svakog pokusa nalazi po jedna čestica. Na svaku komoru se nastavlja **točno jedna** komora. Svake sekunde se sve čestice prebace iz komore u kojoj se trenutno nalaze u komoru koja se nju nastavlja. Primijetite da ako se na komoru A nastavlja komora B, nije nužno da na komoru B nastavlja komora A, iako je i to moguće. Za pokuse je izuzetno važno je da **nakon K sekundi sve čestice budu u svojim početnim komorama**.

Mirka zanima na koliko načina može napraviti nacrt za ubrzivač sa traženim svojstvima. Dva nacrtu se razlikuju ukoliko postoji komora na koju se u ta dva nacrtu nastavljaju različite komore. Kako nacrtu može biti jako mnogo, potrebno je odrediti samo **ostatak pri dijeljenju broja nacrtu sa modulom M**.

**Napomena:** Komora se može nastavljati na samu sebe.

### Ulazni podaci

U prvom retku nalaze se brojevi N i K odvojeni razmakom,  $1 \leq K \leq N \leq 30\,000$ . Broj N predstavlja ukupan broj komora koje Mirko ima na raspolaganju za gradnju ubrzivača, a K je broj sekundi nakon kojeg sve čestice moraju biti u svojim početnim komorama.

U drugom retku nalazi se broj M,  $1 \leq M \leq 10^9$ , modul.

### Izlazni podaci

U prvi i jedini redak izlaza potrebno je ispisati ostatak pri dijeljenju broja različitih nacrtu s brojem M.

### Primjeri test podataka

<b>ulaz</b> 2 1 1000	<b>ulaz</b> 3 2 1000	<b>ulaz</b> 4 4 6
<b>izlaz</b> 1	<b>izlaz</b> 4	<b>izlaz</b> 4

Pojašnjenje drugog primjera: Postoje 4 različita nacrtu, u svakom stupcu donje tablice prikazan je jedan. Oznaka A->B znači da se na komoru A nastavlja komora B.

1 -> 1	1 -> 2	1 -> 3	1 -> 1
2 -> 2	2 -> 1	2 -> 2	2 -> 3
3 -> 3	3 -> 3	3 -> 1	3 -> 2

Pojašnjenje trećeg primjera: Postoji 16 različitih nacrtu, ali traži se ostatak tog broja sa 6. Budući da je  $16 = 2 * 6 + 4$ , potrebno je ispisati samo broj 4.

Mirko i Slavko igraju jednu jako zanimljivu igru. Mirko na (jako dugom) papiru napiše  $N$  brojeva. Slavko tada uzima olovku i prekriži neke od tih brojeva. Nakon toga se svi neprekiženi brojevi prepisuju na novi papir **u istom poretku**, a Slavko dobiva bodove tako da zbroji sve umnoške dvaju susjednih brojeva. Na primjer: ako je prepisani niz  $\{ 3, 6, -1, 2 \}$ , onda Slavko dobiva  $3*6 - 6*1 - 1*2 = 10$  bodova. Ukoliko prepisani niz ima manje od 2 broja, Slavko dobiva **nula bodova**.

Slavko je u nedoumici: ne zna koje brojeve treba prekrižiti na Mirkovom popisu tako da ostvari maksimalni mogući broj bodova. Napišite program će odrediti taj maksimalni mogući broj bodova.

### Ulazni podaci

U prvom redu se nalazi broj  $N$  ( $3 \leq N \leq 30\,000$ ), koliko ima brojeva zapisanih na Mirkovom papiru. U svakom od sljedećih  $N$  redova nalazi se po jedan realan broj  $X_i$  ( $-10.00 \leq X_i \leq 10.00$ ) sa Mirkovog papira, zapisan **na točno dvije decimale, sa nulama na kraju ako je potrebno**.

### Izlazni podaci

U prvi i jedini redak ispišite maksimalni broj bodova koje Slavko može dobiti. Rezultat je potrebno ispsati pomoću **točno četiri** decimalne znamenke.

### Primjeri test podataka

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
3	5	3
8.52	1.00	-1.25
1.00	2.00	10.00
9.61	3.00	-1.00
	4.00	
<b>izlaz</b>	5.00	<b>izlaz</b>
81.8772		1.2500
	<b>izlaz</b>	
	40.0000	

Farmer Mirko se počeo baviti uzgojem krava. On posjeduje  $N$  štala koje se nalaze na **dugoj, ravnoj cesti**, označenih brojevima od 1 do  $N$ . Svakog jutra sve štale treba **otvoriti**, a navečer ih treba sve **zatvoriti**. U Mirkovoju kući nalazi se  $M$  prekidača, označenih brojevima od 1 do  $M$ . Pritiskom na neki prekidač vrata nekih štala se otvore, nekih zatvore, dok **ostala ne mijenjaju svoje stanje**. Ukoliko prekidač pokuša otvoriti (odnosno zatvoriti) neka vrata koja su već otvorena (odnosno zatvorena), onda ta vrata također ne mijenjaju stanje.

Nakon što je stisnuo sve prekidače koje je želio, redosljedom koji je odabrao. Mirko mora pješice obići sve štale koje su zatvorene (odnosno otvorene ako se radi o večeri), ručno ih dovesti u željeno stanje te se, na kraju, **vratiti u svoju kuću**. Naravno, i ujutro i navečer, Mirko želi **propješaćiti najmanju potrebnu udaljenost**.

Pozicije štala su zadane udaljenošću od Mirkove kuće u metrima. Pritom, negativna udaljenost označava štalu koja se nalazi lijevo od Mirkove kuće, a pozitivna štalu koja se nalazi desno. Neka je, na primjer, pozicija prve štale je  $-2$ , druge je  $3$ , treće je  $4$ , a četvrte je  $10$ . Pretpostavimo da Mirko posjeduje sljedeće prekidače:

	1. štala [ -2 ]	2. štala [ 3 ]	3. štala [ 4 ]	4. štala [ 10 ]
1. prekidač	otvara	-	-	zatvara
2. prekidač	zatvara	zatvara	zatvara	otvara
3. prekidač	otvara	otvara	otvara	zatvara

Ujutro se Mirku najviše isplati pristisnuti 2. prekidač te tako otvoriti 4. štalu i zatim obići sve ostale pješice. Tako ukupna propješaćena udaljenost postaje  $2$  (do 1. štale)  $+ 5$  (od 1. do 2.)  $+ 1$  (od 2. do 3.)  $+ 4$  (od 3. do kuće)  $= 12$  metara. Navečer je najisplativije pritisnuti 2. prekidač, a zatim 1. prekidač, što ostavlja otvorenom samo štalu 1. Da bi i nju zatvorio, Mirko treba propješaćiti  $2 + 2 = 4$  metra.

Napišite program koji na temelju pozicija štala i informacija o tome koji prekidač otvara/zatvara koju štalu utvrđuje minimalne moguće udaljenosti koje Mirko treba propješaćiti ujutro, odnosno navečer.

### Ulazni podaci

U prvom retku nalazi se broj štala  $N$  ( $1 \leq N \leq 500$ ), te  $M$  ( $1 \leq M \leq 20\,000$ ) broj prekidača. U sljedećem retku nalazi se  $N$  cijelih brojeva  $X_i$  ( $1 \leq |X_i| \leq 1\,000\,000$ ), pozicije štala zadane **u uzlaznom poretku**.

U idućem retku nalazi se prirodni broj  $G$  ( $0 \leq G \leq 20\,000$ ), broj pravila. Jedno pravilo definira akciju na određenim vratima kada se pritisne određeni prekidač. U sljedećih  $G$  redaka zapisana su pravila u obliku „ $P V T$ “ ( $1 \leq P \leq M$ ,  $1 \leq V \leq N$ ), gdje je  $P$  redni broj prekidača,  $V$  redni broj štale, a  $T$  jedna od riječi „otvara“ ili „zatvara“ – akcija koja se izvrši na vratima štale  $V$  kada se pritisne prekidač  $P$ .

**Napomena:** Neće postojati dvije štale na istim pozicijama. Niti jedne dvije različite akcije neće imati isti uređeni par  $(P, V)$ .

### Izlazni podaci

U prvi redak ispišite minimalnu udaljenost koju Mirko treba pješaćiti ujutro, a u drugi redak minimalnu udaljenost koju treba pješaćiti navečer.

### Primjeri test podataka

<b>ulaz</b>	<b>ulaz</b>
4 3	3 20000
-2 3 4 10	1 2 3
10	3
1 1 otvara	1 1 otvara
1 4 zatvara	1 2 otvara
2 1 zatvara	1 3 otvara
2 2 zatvara	
2 3 zatvara	<b>izlaz</b>
2 4 otvara	0
3 1 otvara	6
3 2 otvara	
3 3 otvara	
3 4 zatvara	
<b>izlaz</b>	
12	
4	

Prvi primjer odgovara tekstu zadatka.

Ljubo je lokalni moćnik malog primorskog mjesta i njegova je odgovornost brinuti za razvoj turizma. Naravno, Ljubo je svjestan da ništa ne privlači turiste više od šetnje prekrasnom betoniranom rivom, pa je stoga odlučio ograditi dio obale kako bi ga mogao zaliti betonom. Kako bi turisti što više uživali, Ljubo želi betonirati dio obale **što je moguće veće površine**.

Obalna crta je zadana izlomljenom linijom koja ne presjeca samu sebe; linija je opisana nizom vrhova  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ , te **uvijek ide sa lijeva na desno**, točnije vrijedi  $x_1 < x_2 < \dots < x_N$ . More se nalazi **iznad**, a obala **ispod** izlomljene linije.

Ljubi je na raspolaganju špaga dugačka  $L$  metara. On će **odabrati dva vrha** izlomljene linije i u njih zabiti dva štapa. Potom će razapeti špagu (ili samo dio špage) između ta dva štapa tako da špaga bude napeta. Dužina koju razapinje špaga **ne smije sjeći more**, ali je dopušteno da **dodiruje obalnu liniju**. Nakon što razapne špagu, Ljubo će betonirati dio obale između špage i mora.



Slika odgovara trećem primjeru. Crna izlomljena linija predstavlja obalnu crtu, more je zasivljeno. Razapeta Ljubina špaga je predstavljena sivom dužinom. Betonirani dio obale je osjenčan.

Napišite program koji, na temelju obalne linije i duljine špage, određuje kolika je **maksimalna površina obale** koju Ljubo može ograditi i zatim izbetonirati. Primjetite da tražena površina može biti i nula.

### Ulazni podaci

U prvom redu ulaza nalaze se cijeli brojevi  $N$  i  $L$ ,  $3 \leq N \leq 5\,000$ ,  $0 \leq L \leq 1\,000\,000$ . Broj  $N$  predstavlja broj vrhova izlomljene linije kojom je opisana obalna crta. Broj  $L$  predstavlja duljinu Ljubine špage.

U svakom od idućih  $N$  redova nalaze se po dva cijela broja  $x_i$  i  $y_i$  odvojena razmakom,  $0 \leq x_i, y_i \leq 1\,000\,000$  — koordinate vrhova obalne linije.

### Izlazni podaci

U jedinom redu izlaza treba ispisati jedan realni broj – najveću moguću površinu obale koju Ljubo može ograditi i betonirati.

Rješenje je potrebno ispisati **sa točno jednom decimalnom znamenkom**. Možete pretpostaviti da će površina uvijek biti ili cijeli broj ili polovina cijelog broja.

### Primjeri test podataka

```
ulaz
5 4
0 0
1 3
2 0
3 3
4 0

izlaz
6.0
```

```
ulaz
3 10
100 100
101 0
102 100

izlaz
0.0
```

```
ulaz
6 5
0 0
2 2
3 1
4 1
6 2
10 0

izlaz
2.5
```

Mirka je počela zanimati kriptografija pa se bacio na izradu sustava javnog ključa baziranom na „Multi-prime RSA“ algoritmu. Detalji sustava nisu bitni za potrebe ovog zadatka, bitno je samo da je važan dio svakog ključa takozvani *modul* – prirodni broj koji je umnožak **dva ili više različitih prostih brojeva**. Za sigurnost sustava nužno je (mada možda ne i dovoljno) da se modul teško može rastaviti na proste faktore u nekom normalnom vremenu.

Kako je Mirko neiskusn, pogriješio je već u prvom koraku – generiraju slučajnih prostih brojeva. Naime, njegov generator nema dovoljno entropije pa može generirati ukupno **samo 36 različitih prostih brojeva**, nadalje svaki generirani prosti broj je **manji od  $10^9$** . Dakle, svaki modul kojeg Mirko generira je umnožak nekih od tih 36 prostih brojeva.

Mirko je generirao  $N$  različitih modula te zamolio prijatelja Slavka da mu pomogne testirati sigurnost tako da proba sve module do kraja rastaviti na proste faktore. Vješti Slavko, koji zna ograničenja Mirkovog generatora, ne samo da će ih sve potpuno rastaviti, nego i želi Mirku dojaviti faktorizaciju na što efikasniji način.

Slavko će Mirku, kao odgovor na njegov izazov, poslati takozvani *dokaz faktorizacije* - **skup prostih brojeva** sa svojstvom da, kada se **svaki modul** podjeli sa **svim prostim brojevima iz dokaza** (sa kojima je djeljiv) kao rezultat se dobije **ili broj jedan ili neki prosti broj**. Kako Mirko zna faktorizaciju, kada primi od Slavka dokaz, može se lako uvjeriti da je Slavko stvarno uspio do kraja rastaviti sve module.

Napišite program koji će na temelju niza od  $N$  različitih modula odrediti kolika je **veličina najmanjeg dokaza faktorizacije**. Primjetite da nije potrebno odrediti same proste brojeve u tom dokazu – samo nas zanima koliko ih je najmanje potrebno.

### Ulazni podaci

U prvom retku nalazi se prirodan broj  $N$  ( $1 \leq N \leq 100\,000$ ), broj modula u Mirkovom izazovu Slavku. U sljedećih  $N$  redaka nalaze se prirodni brojevi  $X_i$  ( $2 \leq X_i \leq 10^{18}$ ), moduli koje Mirko šalje Slavku.

Svaki  $X_i$  je umnožak dva ili više različitih prostih brojeva manjih od  $10^9$ . Ukupan broj prostih brojeva koji se pojavljuju kao faktori je najviše 36.

### Izlazni podaci

U prvi i jedini redak izlaza potrebno je ispisati veličinu najmanjeg dokaza faktorizacije.

### Primjeri test podataka

<b>ulaz</b> 2 35 77	<b>ulaz</b> 3 6 35 77	<b>ulaz</b> 1 999992936000441063
<b>izlaz</b> 1	<b>izlaz</b> 2	<b>izlaz</b> 1

Pojašnjenje prvog primjera:  $35 = 5 \cdot 7$ ,  $77 = 7 \cdot 11$ . Lako je primjetiti da je potreban samo jedan prosti broj kao dokaz faktorizacije, i to broj 7. Dijeljenjem s njime, ulazni moduli se pretvaraju u  $\{5, 11\}$  što su oba prosti brojevi.

Nakon duljeg razmišljanja i traženja smisla života, mali Adrijan je odlučio postati umjetnik. Nažalost, nakon dugogodišnjeg uživanja u matematičkim porocima, kao svoje prvo umjetničko djelo naslikao je primjerak **trokuta**. No jako se izenadio kada mu ravnatelj galerije, stanoviti Mirko, nije želio izložiti sliku. Mислеći kako je problem u nedostatku detalja, mali Adrijan sada želi ukrasiti sliku te ponovno isprobati svoju sreću u umjetničkim galerijama.

Adrijan će ukrasiti sliku tako da na početku odabere neki prirodni broj  $N$ , te trokut kojeg je originalno nacrtao proglasi zanimljivim.

Nakon toga Adrijan  $N-1$  puta ponovi sljedeći postupak: svakom trokutu kojeg je proglasio zanimljivim on spoji polovišta stranica. Na taj će način svaki zanimljivi trokut biti podijeljen na četiri manja trokuta: tri rubna i jedan središnji. Adrijan će zatim zanimljivim proglasiti sve rubne trokute.

Vaš zadatak je pomoći malom Adrijanu tako da napravite simulaciju izgleda njegovog konačnog remek dijela. Za detalje oko formata slike pogledajte izlazne podatke i priložene test primjere.

### Ulazni podaci

U prvom i jedinom redu se nalazi prirodni broj  $N$  ( $1 \leq N \leq 10$ ) iz teksta zadatka.

### Izlazni podaci

Potrebno je nacrtati Adrijanovu sliku nakon ukrašavanja.

Napomene:

- Jedinim znakovima koje smijete koristiti u izlazu su: '/' (slash), '\' (backslash), '\_' (podvlaka), te '.' (točka).
- Izlaz vašeg programa mora točno odgovarati donjim primjerima. Najmanji trokut na izlazu mora sadržavati **točno dva znaka na svojim „kosim“ stranicama** i **točno 4 znaka na horizontalnoj stranici** (tj. treba biti točno onih dimenzija kao rješenje za  $N = 1$ ).
- Nemojte ostavljati višak razmaka (ili točaka) na kraju bilo kojeg retka. Dakle, **zadnji znak u svakom retku mora biti '\'.**
- Nemojte ostavljati prazne redove.

### Primjeri test podataka

```
ulaz
1

izlaz
./\
/__\
```

```
ulaz
2

izlaz
.../\
../__\
./\..\
/__\/__\
```

```
ulaz
3

izlaz
...../\
...../__\
...../\..\
...../__\/__\
...../\...../\
../__\...../__\
./\..\..\..\
/__\/__\/__\/__\
```

Po ploči sastavljenoj od  $N \times N$  jediničnih kvadratića gmižu zmije. Svaka zmija zauzima niz od **barem dva** kvadratića, takav da svaka dva uzastopna kvadratića imaju **zajedničku stranicu**. Prvi kvadratić u nizu nazivamo **glava**. Na svakom kvadratiću ploče se može nalaziti maksimalno jedna zmija. Kvadratići koje na početku ne zauzimaju zmije mogu biti ili prazni ili se u njima može nalaziti prepreka.

Zmija gmiže tako da, u svakom koraku, najprije pomakne glavu na neko **prazno susjedno** polje, a zatim povuče rep ispraznivši tako jedno polje. Smjer kretanja je na početku određen položajem glave u odnosu na drugi po redu kvadratić koji čini zmiju. U svakom koraku zmija se ponaša prema sljedećim pravilima:

- Ukoliko se može pomaknuti naprijed (bez da se pritom sudari sa preprekom, drugom zmijom, samom sobom ili da izađe izvan granica ploče), onda se pomakne naprijed;
- ako je to nemoguće onda **pokuša skrenuti prema desno**;
- ako je to nemoguće onda **pokuša skrenuti prema lijevo**;
- ako je to nemoguće onda **ostane stajati na mjestu**, te se u sljedećem koraku opet pokušava pomaknuti naprijed.

Na ploči se nalazi nekoliko zmija označenih slovima engleske abecede. U svakom koraku sve se pokušavaju se pomaknuti prema gore opisanim pravilima i to **abecednim redosljedom**. Svaki korak traje točno jednu sekundu. Napišite program koji će, za zadani početni položaj zmija na ploči, utvrditi njihov **položaj** nakon što **protekne T sekundi**.

### Ulazni podaci

U prvom redu ulaza nalaze se cijeli brojevi  $N$  i  $T$ ,  $2 \leq N \leq 1000$ ,  $1 \leq T \leq 1000000$ . Broj  $N$  opisuje dimenziju ploče, a  $T$  je broj sekundi nakon kojih treba utvrditi položaj zmija.

U svakom od idućih  $N$  redova nalazi se po  $N$  znakova; ti redovi opisuju početni položaj zmija, praznih kvadratića i prepreka na ploči.

Preciznije, svaki znak može biti:

- ' . ' (točka) – pripadni kvadratić je prazan
- ' # ' – na pripadnom kvadratiću je prepreka
- veliko slovo engl. abecede – na pripadnom kvadratiću je glava zmije
- malo slovo engl. abecede – na pripadnom kvadratiću je dio zmije koji nije glava

Svi kvadratići u kojima se nalazi isto slovo abecede (bilo veliko ili malo) čine jednu zmiju. Pri tome će ulazni podaci biti takvi da će svaki kvadratić koji čini neku zmiju biti susjedan **točno sa dva** druga njena kvadratića, osim glave i vrha repa koji će biti susjedni **točno jednom** njezinom kvadratiću. Naravno, neće postojati dvije različite zmije označene istim slovom.

### Izlazni podaci

Potrebno je ispisati N redova, svaki sa po N znakova. Ti redovi trebaju opisivati položaj zmija po isteku T sekundi, u istom formatu kako je položaj bio opisan u ulaznim podacima.

### Primjeri test podataka

**ulaz**

```
4 8
.bB.
....
a.#.
aaA.
```

**izlaz**

```
.baa
.BAa
..#.
....
```

**ulaz**

```
7 100000
aA.....
a#####D
aa.....d
.....d
.....d
.....d
.....d
```

**izlaz**

```
aaaaADd
.#####d
.....d
.....d
.....d
.....d
.....d
```