

ZADATAK	KAMEN	TIPKE	ASCII
izvorni kôd	kamen.pas kamen.c kamen.cpp kamen.cxx	tipke.pas tipke.c tipke.cpp tipke.cxx	ascii.pas ascii.c ascii.cpp ascii.cxx
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		
memorijsko ograničenje	256 MB		
broj bodova	50	70	80
	200		



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

Mirko je kao mali volio igru Boulder Dash u kojoj junak trči kroz podzemne hodnike sakupljajući dijamante i izbjegavajući kamenje. Vaš zadatak je da napišete program koji simulira dio te igre - padanje kamenja.

Ploča za igru je kvadratna mreža koja se sastoji od R redaka označenih brojevima od 1 do R odozgo prema dolje te S stupaca označenih brojevima od 1 do S s lijeva na desno. Svaki jedinični kvadratić u mreži je ili prazan (označen znakom '.'), ili na njemu može biti prepreka (označena znakom '#') ili kamen (označen malim slovom 'o'). Na početku igre ne postoji niti jedan kamen na ploči.

U svakom koraku se u neki stupac ubacuje točno jedan kamen koji se najprije pojavi u prvom retku tog stupca te dalje pada dolje kroz ploču prema sljedećim pravilima:

- Ako je polje neposredno ispod kamena **prazno**, onda kamen pada na to prazno polje.
- Ako se na polju neposredno ispod kamena **nalazi prepreka** ili se kamen nalazi u **najdonjem redu** ploče, onda kamen ostaje na mjestu te je korak gotov.
- Ako se neposredno ispod kamena nalazi **drugi kamen**, tada se događa sljedeće:
 - Ako su polja neposredno desno i dolje-desno od kamena koji pada prazna (te unutar ploče za igru), tada kamen propada na polje dolje-desno
 - Inače, ako su polja neposredno lijevo i dolje-lijevo od kamena koji pada prazna (te unutar ploče za igru), tada kamen propada na polje dolje-lijevo
 - Inače, kamen ostaje na mjestu te je korak gotov.

```
.....    ...1..
...o...    ...o2.
...#...    ...#3.
.....    ....4.
...ooo    ...5oo
```

Slika 1: Primjer padanja kamena kroz ploču

Na primjer, ako je trenutno stanje ploče kao na slici gore lijevo te kamen ubacujemo u četvrti stupac onda će on padati putanjom naznačenom brojevima na slici desno.

Napišite program koji će na temelju početnog izgleda ploče te zadanog niza koraka gdje je svaki korak ubacivanje kamena u jedan stupac, odrediti **konačni izgled ploče nakon što je svo kamenje ubačeno**.

ULAZNI PODACI

U prvom redu nalaze se dva prirodna broja R i S ($1 \leq R, S \leq 100$), broj redaka i stupaca mreže kvadratića. U svakom od sljedećih R redova nalazi se niz od točno S znakova '.' (točka), '#' - izgled ploče na početku igre.

U sljedećem redu nalazi se prirodni broj K ($1 \leq K \leq 100$), broj kamenja koje ubacujemo u ploču.

U sljedećem redu nalazi se K prirodnih brojeva između 1 i S (uključivo) odvojenih jednim razmakom, redni brojevi stupaca u koje (u zadanom poretku) treba ubaciti kamenje.

Ulaz će biti takav da će uvijek u ploči biti mjesta za kamen koji se ubacuje. Drugim riječima, ako u nekom koraku treba ubaciti kamen u stupac A , tada će u tom trenutku prvi redak stupca A biti prazan.

IZLAZNI PODACI

Potrebno je ispisati R redova, u svaki red točno S znakova - izgled ploče za igru nakon što je svo kamenje ubačeno.

Ploča treba biti identična onoj s ulaza osim što na pozicijama na kojima se nalazi kamenje treba pisati malo slovo 'o'.

BODOVANJE

U 30% test podataka će vrijediti da su svi retci u koje se ubacuje kamenje različiti.

PRIMJERI TEST PODATAKA

ulaz 5 7 #..... .#.#...# 6 2 2 2 2 6 6 izlaz O..... .#O..... .#O#...# ..O.OO.	ulaz 5 6 #.. 7 4 4 4 4 3 4 5 izlaz O.. ...#.. ...OO ..OOOO
--	---

Pojavom tableta i mobitela s ekranima osjetljivima na dodir, fizičke tipkovnice lagano odlaze u zaborav. Međutim, jedan problem s tipkovicama na ekranima je veća mogućnost pogreške prilikom tipkanja, posebno kada je riječ o relativno malenim ekranima.

Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	
	Z	X	C	V	B	N	M		

Slika 2: Raspored slova na tipkovnici

Za ovu priliku pretpostavimo da se tipkovnica sastoji od skupa pravokutnika u standardnom koordinatnom sustavu (koordinata X raste sa lijeva na desno, koordinata Y raste odozdo prema gore) kao na slici gore. Tipkovnica je definirana na sljedeći način:

- Svaki pravokutnik je širine 300 te visine 200.
- Između svaka dva pravokutnika u istom 'redu' se nalazi horizontalni razmak od točno 20.
- Između dva susjedna reda je vertikalni razmak od točno 20.
- Donji lijevi kut pravokutnika koji sadrži slovo 'Q' je na koordinatama (0, 440).
- Donji lijevi kut pravokutnika koji sadrži slovo 'A' je na koordinatama (150, 220).
- Donji lijevi kut pravokutnika koji sadrži slovo 'Z' je na koordinatama (470, 0).

Na temelju ovih pravila i rasporeda tipaka danog u slici gore, mogu se odrediti koordinate pravokutnika za svako slovo. Tako su, na primjer, donji-lijevi i gornji-desni kutovi pravokutnika koji sadrži slovo 'J' jednaki (2070, 220) i (2370, 420) redom. *Središte* pravokutnika definiramo kao polovište dužine između dva suprotna kuta pravokutnika. Tako je, na primjer, središte pravokutnika koji sadrži slovo 'J' točka (2220, 320). Primijetite da kutovi svih pravokutnika imaju parne X i Y koordinate, pa stoga središte svakog pravokutnika ima cjelobrojne koordinate.

Zadan je niz od N točaka u koordinatnom sustavu - to su koordinate gdje je korisnik redom dodirnuo ekran. Za svaki dodir kažemo da je korisnik *stisnuo slovo* ako se točka nalazi **unutar ili na rubu odgovarajućeg pravokutnika**.

Podzadatak A. Napiši program koji će za svaku od N točaka odrediti koje je slovo korisnik stisnuo te ispisati ili nađeno slovo ili znak '?' ako točka ne odgovara niti jednom slovu.

Podzadatak B. Dodatno, zadan je rječnik koji se sastoji od M riječi, a svaka riječ od točno N velikih slova engleske abecede. Vaš program treba odrediti koja je riječ iz rječnika *najbliža* zadanom nizu točaka.

Udaljenost riječi i niza točaka se definira kao suma udaljenosti i -te točke do i -tog slova u riječi za sve i između 1 i N . Udaljenost slova do točke se definira kao kvadrat euklidske udaljenosti između točke i središta pravokutnika koji odgovara tom slovu. Drugim riječima, ukoliko je stisnuta točka (X_1, Y_1) , a središte pravokutnika nekog slova se nalazi na koordinatama (X_2, Y_2) , onda je udaljenost točke od tog slova jednaka $(X_1 - X_2)^2 + (Y_1 - Y_2)^2$.

Ukoliko ima više najbližih riječi, potrebno je ispisati onu koja se **ranije pojavljuje u rječniku**.

ULAZNI PODACI

U prvom redu nalaze se dva prirodna broja, N i M ($N \leq 10$, $M \leq 100$) međusobno odvojena razmakom - broj točaka i broj riječi u rječniku.

Nakon toga slijedi N redova od kojih svaki sadrži dva cijela broja X , Y ($-5000 \leq X, Y \leq 5000$) - koordinate odgovarajuće točke.

Nakon toga slijedi M redova, svaki sadrži jednu riječ **od točno N velikih slova** engleske abecede. Sve riječi će biti različite i bit će poredane abecedno.

IZLAZNI PODACI

U prvi red potrebno je ispisati niz od N znakova (bez razmaka između pojedinih znakova) gdje je svaki znak ili veliko slovo engleske abecede ili znak '?' - rješenje podzadatka A.

U drugi red izlaza potrebno je ispisati jednu od riječi iz rječnika - rješenje podzadatka B.

BODOVANJE

Ukoliko je prvi red ispravan, natjecatelj dobiva 40% bodova za određeni test podatak, čak i ako drugi red neispravan ili nije uopće ispisan.

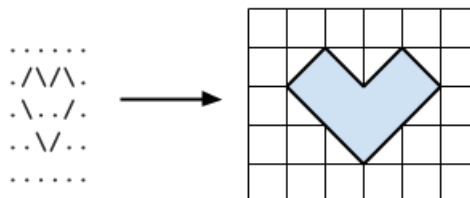
Ukoliko je drugi red ispravan, natjecatelj dobiva 60% bodova za određeni test podatak, čak i ako je prvi red neispravan (međutim, prvi red mora biti ispisan u ispravnom formatu).

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
6 1	4 5	5 6
2070 220	2451 364	912 334
2069 220	2262 433	306 358
2070 219	2202 154	462 349
2370 420	2616 517	-102 340
2371 420	JOZO	770 200
2370 421	KINO	ASADQ
SLAVKO	KRIO	DAASA
izlaz	MIRO	DANAS
J??J??	RIKO	QAZAD
SLAVKO	izlaz	SADAS
	K?NO	SDEFE
	KINO	izlaz
		DA??Z
		SADAS

Mirko je na poklon od Slavka dobio tlocrt Slavkovog voćnjaka i to u obliku datoteke koja sadrži opis tlocrta pomoću ASCII znakova.

Slavkov voćnjak je (ne nužno konveksni) poligon koji je nacrtan na mreži od $R \times S$ jediničnih kvadratića organiziranih u R redaka i S stupaca. Znak '/' označava da postoji zid koji spaja donji lijevi i gornji desni kut kvadratića, znak '\' označava da postoji zid koji spaja gornji lijevi i donji desni kut kvadratića, znak '.' (točka) označava kvadratiće koji ne sadrže u svojoj unutrašnjosti nikakve zidove.



Slika 3: Primjer tlocrta voćnjaka i njegovog opisa pomoću ASCII znakova

Dodatno, **unutar poligona** se mogu nalaziti prazni kvadratići ili **proizvoljni pregradni zidovi** (također označeni kosim crtama '/' i '\'). Međutim, **izvan poligona** se samo mogu nalaziti **prazni kvadratići**.

Napišite program koji će na temelju tlocrta odrediti površinu Slavkovog voćnjaka odnosno poligona.

ULAZNI PODACI

U prvom redu nalaze se dva prirodna broja R i S ($1 \leq R, S \leq 100$), broj redaka i stupaca mreže kvadratića. U sljedećih R redova nalazi se niz od točno S znakova '.', '/' i '\'. Na mreži će se nalaziti točno jedan poligon, koji će možda u svojoj unutrašnjosti sadržavati pregradne zidove.

IZLAZNI PODACI

U prvi i jedini red potrebno je ispisati jedan prirodni broj - površinu poligona. Moguće je dokazati da je površina ovako zadanog poligona uvijek prirodni broj.

BODOVANJE

U 50% test podataka će vrijediti da svi znakovi '/' i '\' koji se pojavljuju pripadaju rubu zadanog poligona odnosno da poligon u svojoj unutrašnjosti **ne sadrži dodatne pregradne zidove**.

PRIMJERI TEST PODATAKA

ulaz 5 6/\./. .\./. .\./. izlaz 6	ulaz 7 10/\..... ..\/.\./\.. ./\/\.\.\.. .\.\./\./\.. ..\/\./\./.. ..\/\./\./.. ..\/\./\./.. ..\/\./\./.. izlaz 24	ulaz 4 4 ../\. ./././. ./././. ././.. izlaz 6
---	--	--