

Baba si ti

Ime zadatka	Baba si ti
Vremensko ograničenje	1 sekunda
Memorijsko ograničenje	1 gigabajt

Anđela igra najnoviji nastavak čuvene franšize "Baba si ti", naslovljen "Soba je mrak".

U ovom izdanju, soba je matrica $H \times W$, gde su redovi numerisani od 0 do $H - 1$ (odozgo na dole), a kolone su numerisane od 0 do $W - 1$ (s leva na desno), što znači da je gornje levo polje $(0, 0)$, a donje desno $(H - 1, W - 1)$. Na jednom od polja, koje nije $(0, 0)$, se nalazi kamen. Sva polja su prohodna, osim kamena, kroz koji se ne može prolaziti. Cela soba je okružena zidovima, kroz koje se takođe ne može prolaziti.

Babom se upravlja tako što joj se zada niz komandi koje predstavljaju smerove u kojima treba da pravi korake. Komande se zadaju stringom koji se sastoji od karaktera "<", ">", "^" i "v". Baba potom izvršava te komande, jednu po jednu. Formalno, ako je Baba na polju (r, c) , "<" predstavlja komandu pomeranja Babe na polje levo, $(r, c - 1)$, ">" na polje desno, $(r, c + 1)$, "^" na polje iznad, $(r - 1, c)$, i "v" na polje ispod, $(r + 1, c)$.

Ako je polje koje je Babi susedno u zadatom smeru prazno, ona prelazi na to polje. Ako to ne može, jer se na njemu nalazi kamen, ili bi udarila u zid, onda nakon te komande Baba ostaje na istom mestu.

Cilj igre je saznati gde je kamen. Međutim, kamen je crn i ne vidi se gde je.

Igra se sastoji od više rundi. U svakoj rundi Baba se postavi poziciju $(0, 0)$ i ugasi se svetlo u sobi. Potom se zadaje niz komandi koje Baba izvršava, ali pošto je mrak, ne vidi se kako se Baba kreće. Na kraju poteza se pali svetlo i igrač saznaje na kom polju je Baba završila izvođenje niza komandi.

Pomozite Anđeli da u što manje rundi sazna gde se nalazi kamen.

Interakcija

Ovo je interaktivni problem.

- Na početku tvoj program treba da pročita liniju koja sadrži dva cela broja, H i W : visinu i širinu sobe.
- Potom, tvoj program treba da komunicira sa grejderom. U svakoj rundi, treba da ispišete znak pitanja "?", nakon koga sledi neprazan string s koji se sastoji od znakova "<", ">", "^", "v". Dužina ovog stringa sme biti najviše 20 000. Potom, tvoj program treba da učitava dva broja r, c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), lokaciju Babe nakon izvršavanja niza komandi. Primeti da se Baba vraća na $(0, 0)$ nakon svake runde.
- Kada možeš da zaključiš gde se nalazi kamen, ispiši "!" prateći ga sa dva broja r_b, c_b , koji predstavljaju red i kolonu kamena ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). Nakon ovoga, tvoj program mora završiti izvršavanje bez pravljenja novih upita. Ovaj poslednji ispis se ne računa kao upit pri bodovanju.

Ne zaboravite da uradite flush standardnog izlaza nakon svakog upita, inače tvoj program može dobiti Time Limit Exceeded presudu. U Python-u, `print()` automatski radi flush. U C++-u, `cout << endl;` takođe radi flush, pored toga što prelazi u novu liniju. Ako koristite `printf`, koristite `fflush(stdout)`.

Grejder je ne-adaptivan, što znači da je pozicija kamena utvrđena pre nego što interakcija počne.

Ograničenja i Bodovanje

- $1 \leq H, W \leq 50$.
- Kamen nikad neće biti lociran na $(0, 0)$. Ovo znači da je $H + W \geq 3$.
- Svaki upit sme da se sastoji od najviše 20 000 instrukcija.
- Možeš izdati najviše 2 500 upita (ispis konačnog rešenja se ne broji u upite).

Tvoje rešenje će biti testirano na više test primera. Ako rešenje padne na *bilo kom* test primeru (npr. tako što ispiše pogrešnu poziciju kamena (Wrong Answer), zakuca se (Runtime Error), prekorači vremensko ograničenje (Time Limit Exceeded), itd.), dobićeš 0 poena i odgovarajuću presudu.

Ako tvoj program uspešno nađe poziciju kamena u *svim* test primerima, dobićeš presudu Accepted, a bodovi će biti izračunati prema sledećoj formuli:

$$\text{rezultat} = \min \left(\frac{100\sqrt{2}}{\sqrt{Q}}, 100 \right) \text{ bodova,}$$

gde je Q najveći broj upita koji je korišćen među svim test primerima. Ispisivanje konačnog rešenja se ne broji u upite. Rezultat će biti zaokružen na najbliži ceo broj.

Dakle, da bi osvojili 100 bodova, program mora rešiti svaki test primer koristeći najviše $Q = 2$ upita. Tabela ispod prikazuje rezultat za pojedine vrednosti Q .

Q	2	3	4	5	...	20	...	50	...	2500
Score	100	82	71	63	...	32	...	20	...	3

Testiranje

Da bi ti pomogli da testiraš svoje rešenje, prilažemo jednostavanu alatku koju možeš preuzeti. Naći ćeš je na Kattis-u pod "attachments" pri dnu stranice problema. Korišćenje ove alatke je opciono i dozvoljeno ti je da je menjaš. Primeti da zvanični grejder program na Kattis-u nije identičan ovoj alatki.

Primer korišćenja (sa $H = 4$, $W = 5$, i skrivenim kamenom na polju $r = 2$, $c = 3$):

Za Python programe, pokrenite `solution.py` (obično se pokreće sa `pypy3 solution.py`):

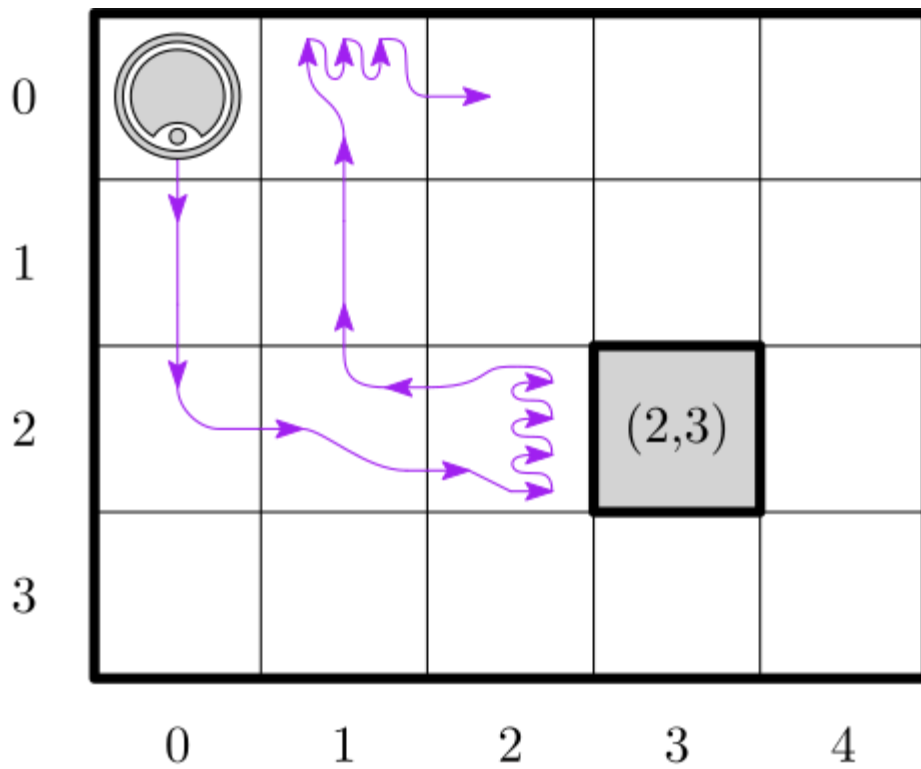
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

Za C++ programe, prvo je kompajliraj (npr. sa `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) a potom pokreni:

```
python3 testing_tool.py ./solution.out <<<"4 5 2 3"
```

Primer

Posmatraj sledeći test primer. Soba ima visinu $H = 4$ i širinu $W = 5$, a kamen je na poziciji $(r, c) = (2, 3)$. Slika ispod prikazuje Babinu putanju pri izvršavanju niza komandi iz prvog upita "`? vv>>>>>><^^^>`", koji rezultuje da Baba završi na poziciji $(r, c) = (0, 2)$. Pre drugog upita, Baba se vraća na gornje levo polje, $(0, 0)$. Nakon toga se zadaje drugi upit "`? >>>>>>>vvvvvvvvvv`", nakon kog se Baba nalazi u donjem desnom ćošku $(r, c) = (3, 4)$. U ovom momentu pogađa se rešenje ispisivanjem "`! 2 3`", što je tačna pozicija kutije.



izlaz grejdera	tvoj izlaz
4 5	
	? w>>>>>>>><^>>>>>>
0 2	
	? >>>>>>>>v v v v v v v v v v
3 4	
	! 2 3