

Najdi škatlo

Naloga	Find the Box
Omejitev časa	1 sekunda
Omejitev spomina	1 gigabyte

Maj je raziskovalka v robotiki, ki dela na LTH. Izvedela je, da se v kleti univerze nahaja vreden zaklad. Zaklad se nahaja v škatli v temni sobi globoko v podzemlju. Žal Maj ne more kar iti in iskati škatle. V kleti je zelo temno in z odhodom tja bi postala sumljiva. Edini način, kako lahko najde zaklad je, da daljinsko nadzira robotski sesalnik, ki se nahaja v kleti.

Klet je predstavljena z mrežo $H \times W$ grid, kjer so vrstice oštevilčene od 0 do $H - 1$ (od zgoraj navzdol) in stolpci z 0 to $W - 1$ (od leve proti desni). Zgornja leva celica je oštevilčena z $(0, 0)$ in desna spodnja z $(H - 1, W - 1)$. Lokacija škatle z zakladom ni poznana. Vsako noč robotski sesalec začne v zgornjem levem kotu in se premika po kleti.

Vsako noč lahko Maj da robotu zaporednje navodil, kako naj se premika po kleti, v obliki stringa iz naslednjih znakov: " $<$ ", " $>$ ", " \wedge " in " \vee ". Če robot stoji na celici (r, c) , ki je z vseh strani neblokirana, " $<$ " premakne robota na celico $(r, c - 1)$, " $>$ " premakne robota na celico $(r, c + 1)$, " \wedge " premakne robota na celico $(r - 1, c)$, in " \vee " premakne robota na celico $(r + 1, c)$.

Stene kleti so trdne, zato se nič ne zgodi, če se robot poskuša premakniti izven mreže. Škatla je tudi trdna in se je ne da premakniti. Na koncu vsake noči robot sporoči svojo lokacijo in se vrne v zgornji levi kot.

Čas je ključen, zato se Maj odloči, da bo poiskala škatlo v čim manj nočeh.

Interakcija

To je interaktivna naloga.

- Tvoj program naj na začetku prebere vrstico z dvema integerjema H in W : višino in širino mreže.
- Tvoj program naj potem interagira z ocenjevalnikom. V vsakem krogu interakcije izpiši vprašaj "?", ki mu sledi ne prazen string s iz znakov " $<$ ", " $>$ ", " \wedge ", " \vee ". Dolžina tega stringa naj bo največ 20 000. Potem naj tvoj program prebere dve celi števili r, c ($0 \leq r \leq H - 1$,

$0 \leq c \leq W - 1$) - lokacijo robota po izvedenem navodilu. Po vsaki izvedeni poizvedbi navodil gre robot vedno nazaj na $(0, 0)$.

- Ko poznaš lokacijo škatle, izpiši "!", ki mu sledita dve celi števili r_b, c_b , vrstica in stolpec, kjer se nahaja škatla ($0 \leq r_b \leq H - 1, 0 \leq c_b \leq W - 1$). Po tem naj se tvoj program zaključi brez dodatnih poizvedb. Ta končni izpis ne šteje kot poizvedba pri določanju ocene tvojemu programu.

Poskrbi, da izplakneš (flush) standardni izhod po zastavljanju vprašanja, sicer bo tvoj program morda ocenjen s prekoračenim časom (Time Limit Exceeded). V Pythonu `print()` izplakuje avtomatično. V C++, `cout << endl;` tudi izplakne skupaj z izpisom nove vrstice, če pa uporabljaš `printf`, uporabi `fflush(stdout)`.

Ocenjevalnik se ne prilagaja - položaj škatle je določen preden se interakcija začne.

Omejitve in ocenjevanje

- $1 \leq H, W \leq 50$.
- Škatla se nikolu ne bo nahajala na $(0, 0)$. To pomeni, da je $H + W \geq 3$.
- Vsaka poizvedba je lahko sestavljena iz največ 20 000 znakov.
- Narediš lahko največ 2 500 poizvedb.

Tvoja rešitev bo testirana na več testnih primerih. Če tvoja rešitev ni uspešna na *kateremkoli* izmed testnih primerov (na primer, napačna lokacija škatle (Wrong Answer), sesutje programa (Runtime Error), prekoračna omejitev časa (Time Limit Exceeded), itd.), bo prejela nič točk.

Če tvoj program uspešno najde položaj škatle za *vse* testne primere, boš prejela oceno AC, točke pa bodo izračunane sledeče:

$$\text{ocena} = \left\lceil \frac{100 \sqrt{2}}{\sqrt{Q}}, 100 \right\rceil \text{ točk},$$

kjer je Q maksimalno število poizvedb na katerem koli testnem primeru. Izpis odgovora ne šteje kot poizvedba. Točke bodo zaokrožene na najbližje celo število.

Natančneje: da prejmeš 100 točk, mora tvoj program rešiti vsak testni primer z uporabo največ $Q = 2$ poizvedb. Tabela prikazuje nekaj vrednosti Q in pripadajoče število točk.

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

Orodje za testiranje

Da ti omogočimo testiranje rešitve, ti ponujamo preprosto orodje za testiranje, ki si ga lahko preneseš. Poglej med priponke ("attachments") na dnu kattis strani z nalogo. Orodje uporabljaš po želji in lahko ga spremeniš. Uradni ocenjevalnik na kattis je drugačen od orodja za testiranje.

Primer uporabe (za $H = 4$, $W = 5$ in škatlo skrito na poziciji $r = 2$, $c = 3$):

Za python program z imenom `solution.py` (običajno ga poženeš z `pypy3 solution.py`):

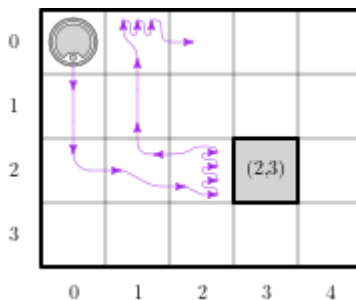
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

C++ programe najprej prevedi (na primer `g++ -std=gnu++17 solution.cpp -o solution.out`) in potem poženi:

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

Primer

Oglej si testni primer. Ta mreža ima višino $H = 4$ in širino $W = 5$, škatla pa je skrita na $(r, c) = (2, 3)$. Slika prikazuje robotovo pot, ko sledi nizu navodil v prvi poizvedbi "`? vv>>>>>><^^^^^>`", ko robot konča na lokaciji $(r, c) = (0, 2)$. Pred drugo poizvedbo gre robot spet nazaj na izhodišče $(0, 0)$. Potem rešitev naredi še eno poizvedbo "`? >>>>>>>vvvvvvvvvv`", pri kateri robot konča na desnem spodnjem kotu $(r, c) = (3, 4)$. Zdaj se rešitev odloči, da ugiba odgovor z izpisom "`! 2 3`", kar je pravilen položaj škatle.



Izhod ocenjevalnika	Tvoj izhod
4 5	
	? vv>>>>>><^^^^^>
0 2	
	? >>>>>>>vvvvvvvvvv
3 4	
	! 2 3