Find the Box

Feladatnév	Find the Box
Időkorlát	1 másodperc
Memóriakorlát	1 gigabyte

Maj a Lundi Egyetem robotikai kutatója. Megtudta, hogy az egyetem pincéjében egy értékes kincs van elrejtve. A kincs egy dobozban van, amely egy üres helyiségben található, mélyen a föld alatt. Sajnos Maj nem tud egyszerűen lemenni és megkeresni a dobozt, mivel nagyon sötét van lent, és ha lámpával menne oda, az gyanút keltene. Az egyetlen módja, hogy megtudja a kincs pontos helyét az, hogy távolról irányít egy robotporszívót, amely a pincében található.

A pincét egy $H \times W$ (hosszúság $\times$ szélesség) cellából álló rácsként ábrázoljuk, ahol a sorok (fentről lefelé) 0-tól $H-1$-ig, és az oszlopok (balról jobbra) 0-tól $W-1$-ig vannak számozva. Ez azt jelenti, hogy a bal felső cella a $(0,0)$, a jobb alsó cella $(H-1, W-1)$ koordinátájú. A kincset tartalmazó doboz egy ismeretlen cellában van, de biztosan nem a $(0,0)$ helyen. A robotporszívó minden este a bal felső sarokból indul, és a kapott utasításnak megfelelően járja be a pincét.

Maj minden este egy sornyi utasítást adhat a robotnak arra, hogy hogyan mozogjon: egy karakterlánc formájában, amiben csak “<”, “>”, “^” és “v” karaktereket használ. Formálisan, ha a robot az (i,j) koordinátájú cellán áll, amely minden oldalról szabad, akkor a "<" balra mozgatja a robotot a $(i, j-1)$ cellába, a ">" karakter a robotot jobbra mozgatja az $(i, j+1)$ koordinátájú cellába, a "^" a robotot felfelé mozgatja az $(i-1, j)$ cellába, és a "v" a robotot lefelé mozgatja az $(i+1, j)$ cellába.

A pince falai szilárdak, így ha a robot megpróbál a rácson kívülre mozogni, semmi sem történik, a robot helyben marad. A kincsesdoboz is szilárd és nem lehet eltolni.

A robot minden éjszaka a mozgás végén jelenti a helyzetét, és visszamegy a bal felső sarokba (a $(0,0)$ koordinátájú cellába).

Az idő sürget, ezért Majnek minél kevesebb éjszaka alatt kell megtalálnia a dobozt.

Interakció

Ez egy interaktív feladat.

- A programodnak egy sor beolvasásával kell kezdenie. Ez a sor két egész számot tartalmaz: H -t és W-t, a rács hosszúságát és szélességét.
- Ezután a programodnak interakcióba kell lépnie az értékelővel. Minden egyes interakciós körben ki kell írnia egy kérdőjelet "`?`", majd egy nem üres, s karakterláncot, amely a "`<`", "`>`", "`^`", "`v`" karakterekből áll. Ennek a karakterláncnak a hossza legfeljebb $20\,000$ lehet.
- Ezután a programodnak két egész számot, r-t és c-t ($0 \leq r \leq H-1,\ 0 \leq c \leq W-1$) kell beolvasnia, a robot helyzetét az utasítások végrehajtása után. Figyelj arra, hogy a robot minden egyes interakció után visszamegy a $(0, 0)$ helyre!
- Ha tudod a kincsesdoboz helyét, írd ki a "`!`" karaktert és utána két egész számot r_b-t és c_b-t, a doboz koordinátáit ($0 \leq r_b \leq H-1,\ 0 \leq c_b \leq W-1$). Ezután a programnak további interakció nélkül ki kell lépnie. Az eredmény kiírása nem számít interakciónak a pontszám meghatározásakor.

Ügyelj arra, hogy a kérdésfeltevés után a standard kimenetet ki kell üríteni, különben a programod időlimit túllépésként (Time Limit Exceeded) értékelhetik. Pythonban a `print()` automatikusan megteszi ezt.

C++-ban a `cout << endl;` is kiürít, egy új sor kiírásával; ha `printf` függvényt használsz, akkor add ki a `fflush(stdout);` parancsot.

Az értékelő nem adaptív, ami azt jelenti, hogy a doboz pozíciója az interakció megkezdése előtt kerül meghatározásra, azaz az interakciók során nem változik.

Korlátok és pontozás

- $1 \leq H, W \leq 50$.
- A kincsesdoboz soha nem lesz a $(0, 0)$ ponton. Ez azt jelenti, hogy $H + W \geq 3$.
- Minden lekérdezés legfeljebb $20\,000$ utasításból állhat.
- Legfeljebb $2\,500$ lekérdezés adható ki. (Az egyes eredmények kiírása nem számít lekérdezésnek.)

A megoldásodat számos teszteseten teszteli az értékelő. Ha a megoldásod bármelyik tesztesetnél hibázik (pl. rossz kincsesdoboz pozíciót ír ki (Wrong Answer), összeomlik (Runtime Error), túllépi az időkorlátot (Time Limit Exceeded) stb.), akkor 0 pontot kapsz.

Ha a programod minden tesztesetben sikeresen megtalálja a kincsesdoboz helyzetét, akkor megkapod az Elfogadva (Accepted) ítéletet, és a következőképpen kiszámított pontszámot:

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

ahol Q a valamelyik tesztesethez használt interakciók maximális száma. A végső válasz, a kincsesláda helyzetének kiírása nem számít interakciónak. A pontszámot a legközelebbi egész számra kerekítjük.

A 100 pont megszerzéséhez a programnak minden tesztesetet legfeljebb $Q = 2$ lekérdezéssel kell megoldania. Az alábbi táblázat a Q néhány értékét és a hozzá tartozó pontszámot mutatja:

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

Tesztelő eszköz

A megoldásod tesztelésének megkönnyítése érdekében egy egyszerű eszközt biztosítunk, amelyet letölthetsz. Ezt a feladatoldal alján, az "attachments" menüpont alatt találod. Az eszköz használata opcionális, és szabadon megváltoztathatod. Vedd figyelembe, hogy a végső értékelő eltér a tesztelő eszköztől.

Példahasználat ($H = 4$, $W = 5$, a kincsesdoboz a $r = 2$, $c = 3$ helyen van):

Pythonban a `solution.py` program (általában `pypy3 solution.py` helyett):

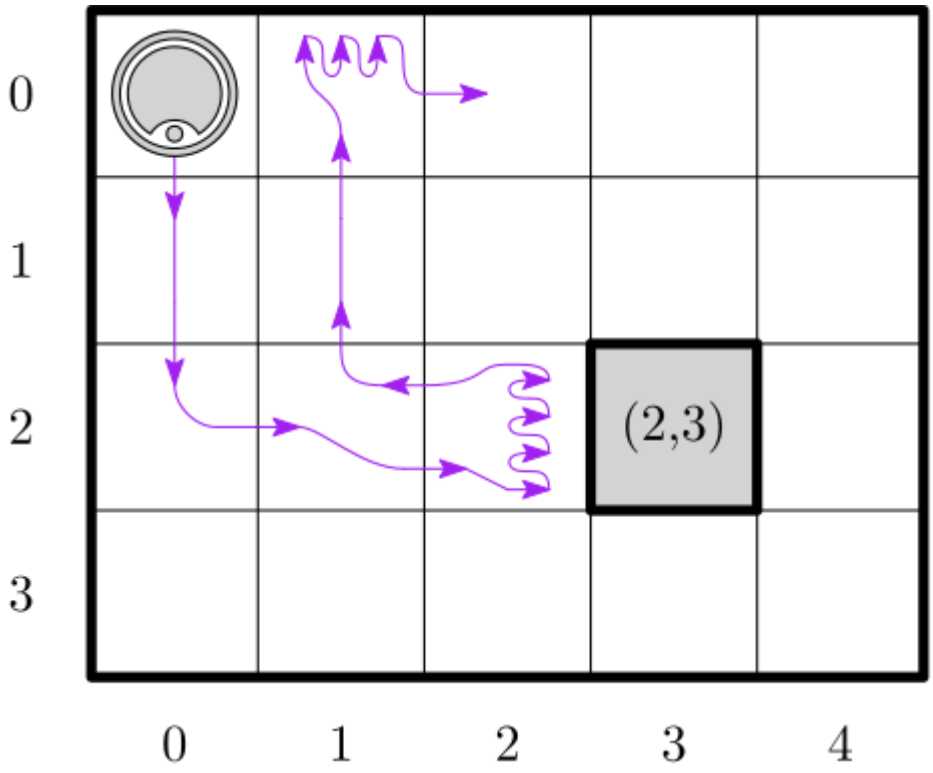
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

C++-ban először fordítsd le a programod (például: `g++ -std=gnu++17 solution.cpp -o solution.out`) és utána futtasd:

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

Példa

Tekintsük a minta tesztesetét. A rács hosszúsága $H = 4$ és szélessége $W = 5$, a kincsesdoboz az $(r, c) = (2, 3)$ pozícióban van. Az alábbi ábra a robot útját mutatja, amikor követi az első "`? vv>>>>>><^^^^^>`" interakció utasításait, amelynek eredményeképpen a robot a $(r, c) = (0, 2)$ pozícióba kerül. A második interakció előtt a robot visszamegy a bal felső sarokba, a $(0, 0)$ koordinátájú cellába. Ezután a megoldás újabb utasítássort ad ki "`? >>>>>>>vvvvvvvvvvvvvvvvv`", amelyre a robot a jobb alsó sarokba, az $(r, c) = (3, 4)$ koordinátájú cellába kerül. Ekkor a programod kiírja a "`! 2 3`" karaktersorozatot, ami a kincsesdoboz helyes helyzetét adja.



értékelő kimenete	a programod kimenete
4 5	
	? vv>>>>>><^^^^^>
0 2	
	? >>>>>>>>vvvvvvvvv
3 4	
	! 2 3