

Find the Box

| Numele problemei | Find the Box |
|-------------------|--------------|
| Limită de timp | 1 second |
| Limită de memorie | 1 gigabyte |

Maj este un cercetător în robotică care lucrează la Universitatea Lund. Ea a aflat despre o comoară valoroasă în pivnița universității. Comoara se află într-o cutie situată într-o cameră goală, adânc sub pământ. Din păcate, Maj nu poate doar să meargă și să caute cutia. Este foarte întunecat în pivniță și dacă ar merge acolo cu o lumină, ar trezi suspiciuni. Singurul ei mod de a găsi comoara este să controleze de la distanță un aspirator robot care locuiește în pivniță.

Pivnița este reprezentată ca un grid de dimensiuni $H \times W$, unde liniile sunt numerotate de la 0 la $H - 1$ (de sus în jos), iar coloanele sunt numerotate de la 0 la $W - 1$ (de la stânga la dreapta), însemnând că celula din colțul stânga sus este $(0, 0)$, iar celula din colțul dreapta jos este $(H - 1, W - 1)$. Cutia cu comoară se află într-o celulă necunoscută, diferită de celula $(0, 0)$. În fiecare noapte, robotul începe din colțul stânga sus și se mișcă prin pivniță.

În fiecare noapte, Maj poate da robotului o secvență de instrucțiuni despre cum ar trebui să se miște sub forma unui șir format din caracterele " $<$ ", " $>$ ", " \wedge " și " \vee ". Formal, dacă robotul se află pe celula (r, c) care este liberă pe toate părțile, " $<$ " mută robotul la stânga către celula $(r, c - 1)$, " $>$ " mută robotul la dreapta către celula $(r, c + 1)$, " \wedge " mută robotul în sus către celula $(r - 1, c)$, și " \vee " mută robotul în jos către celula $(r + 1, c)$.

Pereții pivniței sunt solizi, deci dacă robotul încearcă să se deplaseze în afara grilei, nu se va întâmpla nimic. Cutia este de asemenea solidă și nu poate fi împinsă. La sfârșitul fiecărei nopți, robotul își va raporta locația și se va întoarce în colțul din stânga sus.

Timpul este esențial, așa că Maj decide să găsească cutia în cât mai puține nopți.

Interacțiune

Aceasta este o problemă interactivă.

- Programul tău ar trebui să înceapă prin citirea unei linii cu două numere întregi H și W : înălțimea și lățimea grilei.

- Apoi, programul tău ar trebui să interacționeze cu graderul. În fiecare rundă de interacțiune, ar trebui să tipărești un semn de întrebare "?", urmat de un șir s nevid format din caracterele "<", ">", "^", "v". Lungimea acestui șir poate fi cel mult 20000. Apoi, programul ar trebui să citească două numere întregi r, c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), locația robotului după executarea instrucțiunilor. Reține că robotul se întoarce întotdeauna la $(0, 0)$ după fiecare interogare.
- Când știi locația cutiei, tipărește "!" urmat de cele două numere întregi r_b, c_b , rândul și coloana cutiei ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). După aceasta, programul tău trebuie să iasă fără a mai face alte interogări. Acest output final nu se numără ca o interogare atunci când se calculează scorul tău.

Nu uita să dai flush la output după fiecare interogare, altfel programul poate lua verdictul Time Limit Exceeded. În Python, `print()` dă flush automat. În C++, `cout << endl;` dă flush în adăuga printării unei noi linii; în cazul folosirii funcției `printf`, folosește `fflush(stdout)`.

Graderul nu este adaptiv, adică poziția cutii este stabilită înainte de începerea interacțiunii.

Restricții și punctaj

- $1 \leq H, W \leq 50$.
- Cutia nu se va afla fi niciodată la poziția $(0, 0)$. Acest lucru înseamnă că $H + W \geq 3$.
- Fiecare query poate conține cel mult 20 000 de instrucțiuni.
- Poți face maxim 2 500 de interogări (afișarea răspunsului final nu se numără ca și interogare).

Soluția voastră va fi testată pe mai multe grupe de teste. Dacă soluția ta eșuează în *oricare* dintre aceste grupe (de exemplu, prin raportarea unei poziții greșite a cutiei (Wrong answer), prin erori de rulare (Runtime Error), depășind limita de timp Time Limit Exceeded), etc.), vei primi 0 puncte și verdictul corespunzător.

Dacă programul tău găsește cu succes poziția cutiei în toate cazurile de test, vei primi verdictul Accepted, și un scor calculat după cum urmează:

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

unde Q este numărul maxim de interogări făcute în oricare grupă de test. Afișarea răspunsului final nu se numără ca și interogare. Scorul va fi rotunjit la cel mai apropiat întreg.

Mai specific, pentru a primi 100 de puncte, programul trebuie să rezolve fiecare grupă de test din cel mult $Q = 2$ interogări. Tabelul de mai jos arată scorul asociat pentru diverse valori ale lui Q .

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

Tool de testare

Pentru a facilita testarea soluției tale, oferim un tool simplu pe care îl poți descărca. Consultă secțiunea "atașamente" din partea de jos a paginii Kattis a problemei . Utilizarea toolului este opțională, iar tu ai permisiunea să îl modifichi. Reține că graderul oficial de pe Kattis este diferit de instrumentul de testare.

Exemplu de utilizare (cu $H = 4$, $W = 5$, și cutia ascunsă la poziția $r = 2$, $c = 3$):

Pentru programele Python, `solution.py` (rulează normal drept `pypy3 solution.py`):

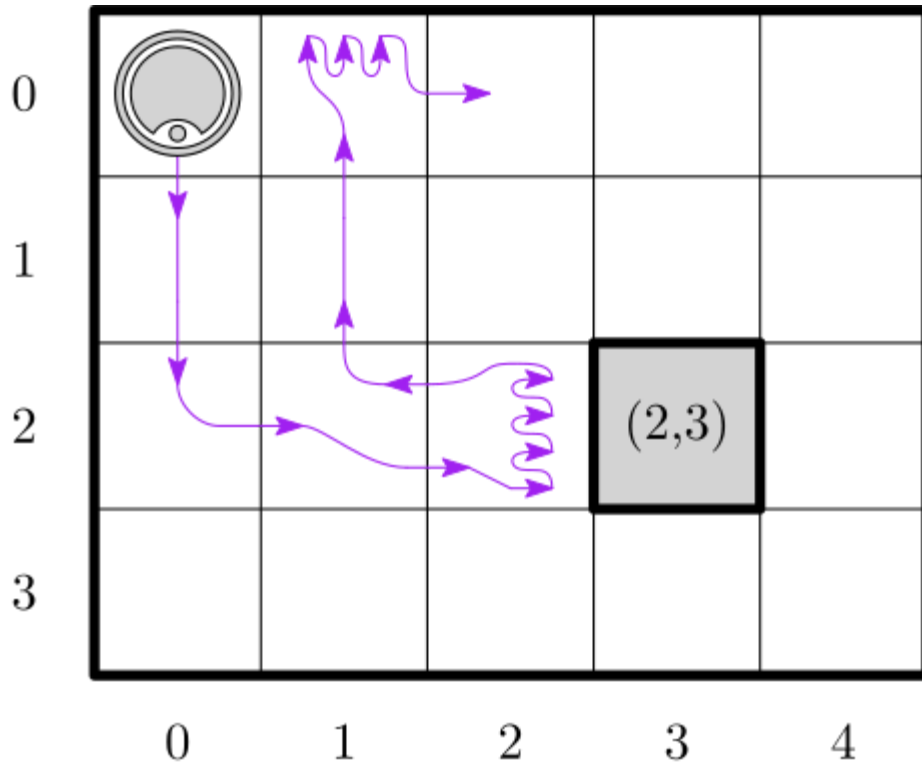
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

Pentru programele C++, mai întâi compilează cu (e.g. with `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) și apoi rulează:

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

Exemplu

Consideră următorul exemplu. Gridul are înălțimea $H = 4$ și lățimea $W = 5$, iar cutia se află la poziția $(r, c) = (2, 3)$. Figura de mai jos arată calea robotului atunci când urmează instrucțiunile primei interogări "`? vv>>>>><^^^^>`", care duce la finalizarea parcurșului robotului la poziția $(r, c) = (0, 2)$. Înainte de a doua interogare, robotul se va întoarce din nou în colțul din stânga sus $(0, 0)$. Apoi, soluția emite o altă interogare "`? >>>>>>>vvvvvvvvvv`" pentru care robotul ajunge în colțul din dreapta jos $(r, c) = (3, 4)$. Acum, soluția decide să ghicească răspunsul, scriind "`! 2 3`", care este poziția corectă a cutiei.



| outputul graderului | outputul tău |
|---------------------|---------------------|
| 4 5 | |
| | ? vv>>>>>>><^>>>>> |
| 0 2 | |
| | ? >>>>>>>>vvvvvvvvv |
| 3 4 | |
| | ! 2 3 |