

Знайдіть коробку

Назва задачі	Find the Box
Обмеження часу	1 с
Обмеження використання пам'яті	1024 МБ

Даша — дослідник робототехніки, який працює в університеті Лунда. Вона дізналася про цінний скарб у підвалі університету. Скарб знаходиться в коробці, розташованій в порожній кімнаті глибоко під землею. На жаль, Даша не може просто піти і шукати коробку. У підвалі дуже темно, і заходити туди зі світлом викликало б підозру. Її єдиний спосіб знайти коробку — дистанційно керувати роботом-пилососом, який живе у підвалі.

Підвал представлений у вигляді сітки $H \times W$, де ряди пронумеровані від 0 до $H - 1$ (зверху вниз) і стовпці пронумеровані від 0 до $W - 1$ (зліва направо), що означає, що верхня ліва клітинка $(0, 0)$ а нижня права клітинка $(H - 1, W - 1)$. Коробка зі скарбом знаходиться в якійсь невідомій клітинці, відмінній від $(0, 0)$. Щовечора робот-пилосос запускається у верхньому лівому куті та переміщається по підвалу.

Кожної ночі Даша може давати роботу послідовність інструкцій про те, як він повинен рухатися, у вигляді рядка, що складається з символів «<», «>», «^» і «v». Формально, якщо робот стоїть на клітинці (r, c) , яка розблокована з усіх боків, «<» переміщує робота вліво до клітинки $(r, c - 1)$, «>» переміщує робота вправо до клітинки $(r, c + 1)$, «^» переміщує робота вгору до клітинки $(r - 1, c)$, а «v» переміщує робота вниз до клітинки $(r + 1, c)$.

Стіни підвалу міцні, тому якщо робот спробує вийти за межі сітки, то нічого не станеться. Коробка також міцна і її неможливо штовхнути. Наприкінці кожної ночі робот повідомлятиме про своє місцезнаходження та повертатиметься у верхній лівий кут.

Час є важливим чинником, тому Даша хоче знайти коробку за якомога меншу кількість ночей. Допоможіть їй у цьому.

Протокол взаємодії

Це є інтерактивна задача.

- Ваша програма повинна починатися з читання рядка з двома цілими числами H і W : висота і ширина сітки.

- Потім ваша програма повинна взаємодіяти з інтерактором. У кожному раунді взаємодії ви повинні надрукувати знак питання «?», а потім непорожній рядок s , який складається з символів «<», «>», «^», «v». Довжина цього рядка може бути щонайбільше 20 000. Потім ваша програма має прочитати два цілих числа r, c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), розташування робота після виконання інструкцій. Зауважте, що робот завжди повертається до $(0, 0)$ після кожного запиту.
- Коли ви дізнаєтесь про розташування коробки, надрукуйте «!», а потім два цілих числа r_b, c_b , рядок і стовець коробки ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). Після цього ваша програма має завершити роботу без подальших запитів. Вивід кінцевого результату не вважається запитом під час визначення вашого балу.

Переконайтесь, що після виводу запиту ви викликаєте функцію `flush` для стандартного виводу, інакше ваша програма отримає `Time Limit Exceeded`. У Python функція `print()` автоматично викликає `flush`. У C++, оператор `cout << endl;` крім виводу нового рядка також викликає `flush`; якщо використовується функція `printf`, використовуйте `fflush(stdout)`.

Перевіряюча система є незмінною (некоригуючою), це означає, що положення коробки визначається до початку взаємодії.

Обмеження та оцінювання

- $1 \leq H, W \leq 50$.
- Коробка ніколи не буде розташована в $(0, 0)$. Це означає, що $H + W \geq 3$.
- Кожен запит може містити щонайбільше 20 000 інструкцій.
- Ви можете надсилати щонайбільше 2 500 запитів (вивід остаточної відповіді не вважається запитом).

Ваше рішення буде протестовано на кількох наборах тестів. Якщо ваше рішення невірне на *будь-якому* з цих тестів (наприклад, повідомляє неправильне положення коробки (`Wrong Answer`), викликає помилку (`Runtime Error`), перевищує ліміт часу (`Time Limit Exceeded`) тощо), ви отримаєте 0 балів та відповідний вердикт.

Якщо ваша програма успішно знаходить положення коробки на *всіх* тестових випадках, ви отримаєте вердикт `Accepted` і бали, які розраховуються наступним чином:

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

де Q — це максимальна кількість запитів виконаних вами серед усіх тестових випадків. Виведення кінцевої відповіді не враховується як запит. Оцінка буде округлена до найближчого цілого числа.

Зокрема, для отримання 100 балів ваша програма повинна вирішити кожен тестовий випадок, використовуючи не більше $Q = 2$ запитів. У таблиці нижче наведені деякі значення Q і відповідні оцінки.

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

Інструмент тестування

Для полегшення тестування вашого рішення ми надаємо простий інструмент, який ви можете завантажити. Перегляньте розділ "Додатки" унизу сторінки задачі на Kattis. Використання цього інструменту є необов'язковим, і ви можете його змінювати. Зверніть увагу, що офіційна програма оцінювача на Kattis відрізняється від цього інструменту тестування.

Приклад використання (з $H = 4$, $W = 5$ і прихованою коробкою в позиції $r = 2$, $c = 3$):

Для програм Python, наприклад, `solution.py` (як правило, запускається як `python3 solution.py`):

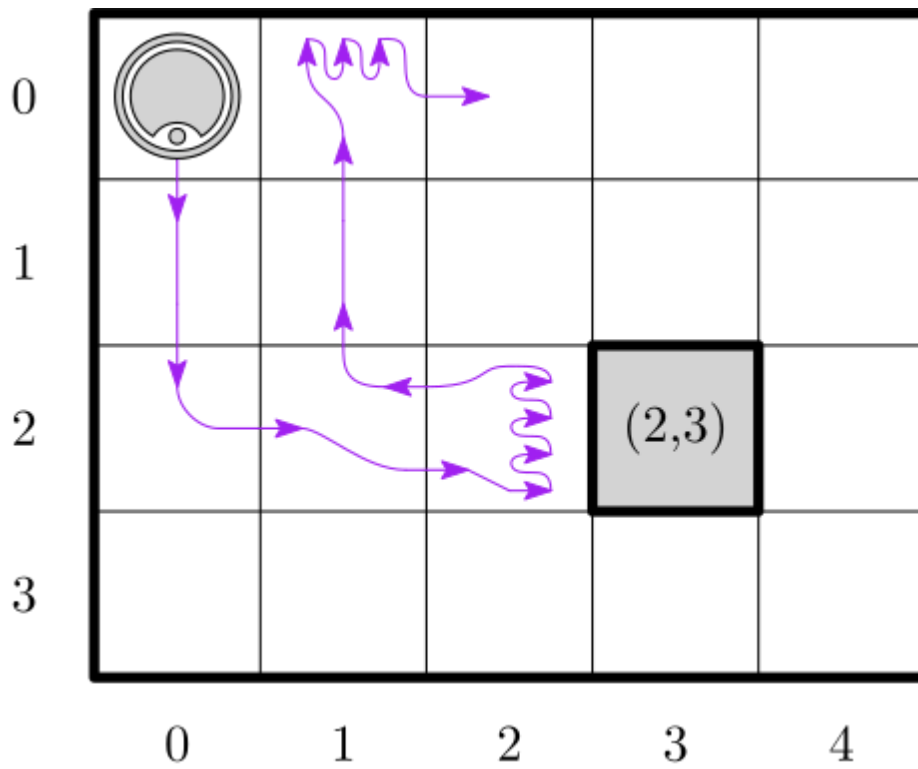
```
python3 testing_tool.py python3 solution.py <<<"4 5 2 3"
```

Для програм C++ спочатку скомпілюйте їх (наприклад, за допомогою `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`), а потім запустіть:

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

Приклад

Розглянемо приклад. Сітка має висоту $H = 4$ і ширину $W = 5$, а коробка знаходиться в позиції $(r, c) = (2, 3)$. На малюнку нижче показано шлях робота під час виконання вказівок першого запиту «? vv>>>>><^^^^^>», в результаті чого робот опиняється в позиції $(r, c) = (0, 2)$. Перед другим запитом робот знову повернеться у верхній лівий кут $(0, 0)$. Потім задається інший запит «? >>>>>>vvvvvvvvvv», після якого робот опиняється в нижньому правому куті $(r, c) = (3, 4)$. Тепер виводимо рішення, написавши «! 2 3», що є правильним положенням коробки.



Вивід інтерактора	Ваш вивід
4 5	
	? vv>>>>>>><^<<<<<<>
0 2	
	? >>>>>>>>vvvvvvvvvvv
3 4	
	! 2 3