

Намерете кутията

Име на задачата	Намерете кутията
Time Limit	1 second
Memory Limit	1 gigabyte

Мая е изследовател по роботика, който работи в университета в Лунд.

Тя е научила за ценно съкровище в мазето на университета.

Съкровището е в кутия, разположена в празна стая дълбоко под земята.

За съжаление Мая не може просто да отиде и да потърси кутията. В мазето е много тъмно и ходенето там с лампа би предизвикало подозрение. Единственият ѝ начин да намери съкровището е да управлява дистанционно прахосмукачка-робот, която обитава мазето.

Избата е представена като $H \times W$ мрежа, където редовете са номерирани от 0 до $H - 1$ (от горе надолу), а колоните са номерирани от 0 до $W - 1$ (от ляво надясно), което означава, че горната лява клетка е $(0,0)$, а долната дясна клетка е $(H - 1, W - 1)$. Кутията със съкровището е в някаква неизвестна клетка, различна от клетката $(0,0)$. Всяка вечер роботът прахосмукачка започва от горния ляв ъгъл и се движи из мазето.

Всяка вечер Мая може да даде на робота последователност от инструкции как трябва да се движи под формата на низ, състоящ се от знаци "<", ">", "^" и "v". Формално, ако роботът се намира в клетка (r, c) , която е проходима във всички посоки, "<" премества робота наляво към клетка $(r, c - 1)$, ">" премества робота надясно до клетка $(r, c + 1)$, "^" премества робота нагоре до клетка $(r - 1, c)$ и "v" премества робота надолу до клетка $(r + 1, c)$.

Стените на мазето са здрави, така че ако роботът се опита да излезе извън мрежата, нищо няма да се случи.

Кутията също е здрава и не може да се бутне. В края на всяка вечер роботът ще съобщава местоположението си и ще се връща в горния ляв ъгъл.

Времето е от съществено значение, така че Мая решава да намери кутията за възможно най-малко нощи.

Взаимодействие

Това е интерактивна задача.

- Вашата програма трябва да започне с четене на ред с две цели числа H и W : височината и ширината на мрежата.
- След това вашата програма трябва да взаимодейства с грейдъра. За всеки ден трябва да отпечатате въпросителен знак „?”, последван от непразен низ s , състоящ се от знаци „<“, „>“, „^“, „v“. Дължината на този низ може да бъде най-много 20 000.

След това вашата програма трябва да прочете две цели числа r и c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), местоположението на робота след изпълнение на инструкциите. Имайте предвид, че роботът винаги се връща в $(0, 0)$ след всеки въпрос.

- Когато знаете местоположението на кутията, отпечатайте „!“, последвано от двете цели числа r_b, c_b , редът и колоната на клетката, в която се намира кутията ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). След това вашата програма трябва да приключи, без да задава допълнителни въпроси. Този краен изход не се брои като въпрос при определяне на вашия резултат.

Уверете се, че сте изчистили стандартния изход след всеки въпрос, или в противен случай вашата програма може да бъде оценена като Time Limit Exceeded.

В Python `print()` изчиства автоматично стандартния изход. В C++, `cout << endl;` също изчиства автоматично стандартния изход, в допълнение към отпечатването на нов ред; ако използвате `printf`, използвайте `fflush(stdout)`.

Ограничения и оценяване

- $1 \leq H, W \leq 50$.
- Кутията никога няма да се намира в клетка $(0, 0)$. Това означава, че $H + W \geq 3$.
- Всяка заявка може да се състои от най-много 20 000 инструкции.
- Можете да задавате най-много 2 500 заявки, (отпечатването на крайния отговор не се брои за заявка).

Вашето решение ще бъде тествано върху няколко теста. Ако вашето решение е неуспешно на *който и да е* от тези тестови случаи (напр. чрез грешна позиция на кутията (WA), срив (RTE), превишаване на времевия лимит (TLE) и т.н.), ще получите 0 точки и подходящото съобщение.

Ако вашата програма успешно намери позицията на кутията за всички тестови случаи, ще получите оценка Accepted и резултат, изчислен както следва:

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

където Q е максималният брой въпроси, използван във всеки тестов случай. Отпечатването на крайния отговор не се брои за въпрос. Резултатът ще бъде закръглен до най-близкото цяло число.

По-точно, за да получите 100 точки, вашата програма трябва да реши всеки тестов случай, използвайки най-много $Q = 2$ въпроса. Таблицата по-долу показва някои стойности на Q и свързания резултат.

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

Инструмент за тестване

За да улесним тестването на вашето решение, ние предоставяме прост инструмент, който можете да изтеглите. Вижте „attachments“ в долната част на страницата със задачата в Kattis. Инструментът не е задължителен за използване и имате право да го промените. Имайте предвид, че официалната програма за оценяване на Kattis е различна от инструмента за тестване.

Примерна употреба (с $H = 4$, $W = 5$ и скрита кутия на позиция $r = 2$, $c = 3$):

За програми на Python напишете `solution.py` (обикновено се изпълнява като `python3 solution.py`):

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

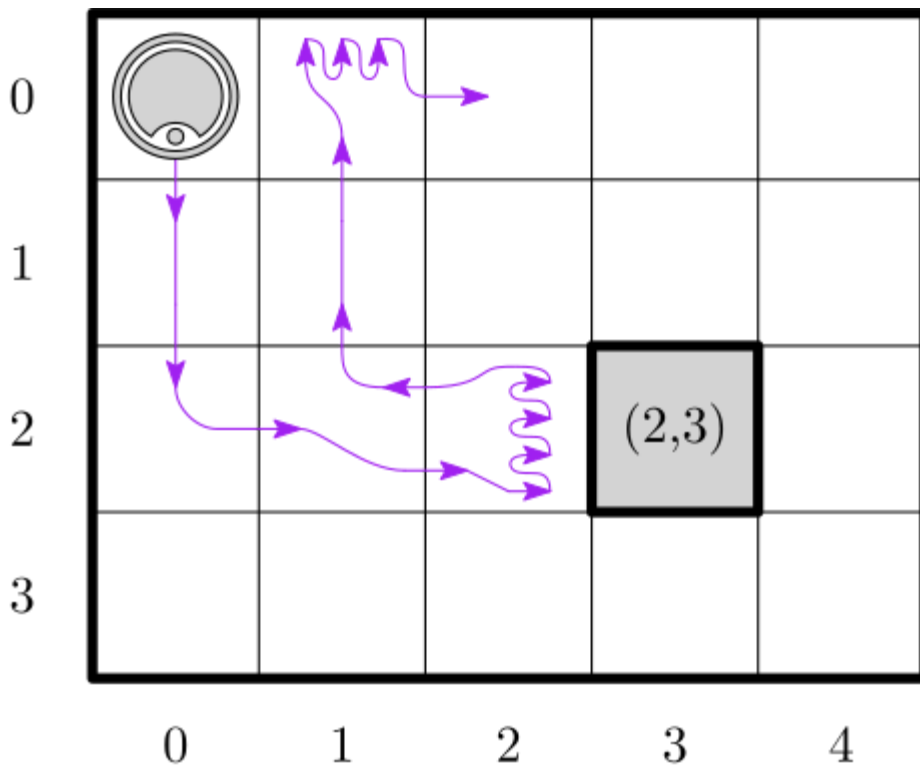
За програми на C++, първо ги компилирайте (т.е. с `g++ -std=gnu++17 solution.cpp -o solution.out`) и след това изпълнете:

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

Пример

Разгледайте примерния тестов случай. Мрежата има височина $H = 4$ и ширина $W = 5$, а кутията е на позиция $(r, c) = (2, 3)$. Фигурата по-долу показва пътя на робота, когато следва инструкциите на първата заявка „? vv>>>>>><^^^^^>“, което води до това, че роботът завършва на позиция $(r, c) = (0, 2)$. Преди второто запитване, роботът ще се върне отново в горния ляв ъгъл $(0, 0)$. Тогава решението издава друга заявка „? >>>>>>>vvvvvvvvvv“, за

което роботът завършва в долния десен ъгъл $(r, c) = (3, 4)$. Сега решението решава да познае отговора, като напише „! 2 3“, което е правилната позиция на кутията.



Изход на грейдъра	Ваш изход
4 5	
	? w>>>>>>><^lllll>
0 2	
	? >>>>>>>vwwwwwww
3 4	
	! 2 3