

## იპოვე ყუთი

ამოცანის სახელი	იპოვე ყუთი
დროის ლიმიტი	1 წამი
მეხსიერების ლიმიტი	1 გიგაბაიტი

მაი რობოტოტექნიკის მკვლევარია და ლუნდის უნივერსიტეტში მუშაობს. მან გაიგო, რომ უნივერსიტეტის სარდაფში ძვირფასი საგანძური ინახება. საგანძური ყუთშია მოთავსებული და ღრმა მინისქვეშეთშია დამალული. სამწუხაროდ, მაის არ შეუძლია უბრალოდ სარდაფში ჩავიდეს და მოძებნოს ეს ყუთი, რადგან იქ ძალიან ბნელა, ხოლო ფარანის წაღება და იქ ანთება ვინმეს ყურადღებას მიიპყრობს და ეჭვებს გამოიწვევს. ამიტომ, საგანძურის პოვნის ერთადერთი საშუალება დისტანციურად მართვადი რობოტის გამოყენებაა.

სარდაფი წარმოდგენილია  $H \times W$  ზომის ბადის სახით, სადაც სტრიქონები გადანომრილია 0-დან  $(H - 1)$ -მდე (ზემოდან ქვემოთ) და სვეტები გადანომრილია 0-დან  $(W - 1)$ -მდე (მარცხნიდან მარჯვნივ), რაც ნიშნავს, რომ ზედა მარცხენა უკრაა  $(0, 0)$  და ქვედა მარჯვენა უკრაა  $(H - 1, W - 1)$ . საგანძურიანი ყუთი რომელიღაც უცნობ უკრაშია მოთავსებული, რომელიც განსხვავებულია  $(0,0)$  უკრისაგან. ყოველ ღამით რობოტი მოძრაობას იწყებს ზედა მარცხენა უკრიდან და სარდაფში გადაადგილდება. ასევე, ყოველ ღამით, მაის შეუძლია რობოტს მისცეს გარკვეული ინსტრუქციების მიმდევრობა, თუ როგორ უნდა იმოძრაოს მან. ინსტრუქციებს ის იძლევა სტრიქონის სახით, რომელიც შეიცავს სიმბოლოებს "<", ">", "^" და "v". ფორმალურად, თუ რობოტი დგას უკრაზე  $(r, c)$ , რომელიც არცერთი მხრიდან ბლოკირებული არ არის, მაშინ ინსტრუქციას "<" ის გადაჰყავს მარცხენა  $(r, c - 1)$  უკრაში, ინსტრუქციას ">" - მარჯვენა  $(r, c + 1)$ , უკრაში, ინსტრუქციას "^" - ზედა  $(r - 1, c)$  უკრაში და ინსტრუქციას "v" - ქვედა  $(r + 1, c)$  უკრაში.

სარდაფის კედლები სქელი და მტკიცეა. ასე, რომ თუ რობოტი ბადის გარეთ გასვლას მოინდომებს, არაფერი არ მოხდება. ყუთიც სქელი და მძიმეა და მისი დაძვრა შეუძლებელია. ყოველი ღამის ბოლოს, რობოტი მაის ატყობინებს თავის ადგილმდებარეობას და ბრუნდება უკან ზედა მარცხენა უკრაში.

დროს გადამწყვეტი მნიშვნელობა აქვს და ამიტომ მაის სურს ყუთი რაც შეიძლება ნაკლები ღამეების განმავლობაში იპოვოს.

## ინტერაქცია

ეს ინტერაქტიული ამოცანაა.

- თქვენმა პროგრამამ უნდა დაინყოს სტრიქონის ნაკითხვით, რომელშიც ორი მთელი  $H$  და  $W$  რიცხვია - ბადის სიმაღლე და სიგანე შესაბამისად.
- შემდეგ, თქვენმა პროგრამამ ინტერაქცია უნდა მოახდინოს გრაფერთან. ინტერაქციის ყოველ რაუნდში თქვენ უნდა დაბეჭდოთ კითხვის ნიშანი "?", ხოლო მას უნდა მოსდევდეს არაცარიელი  $s$  სტრიქონი, რომელიც შეიცავს სიმბოლოებს "<", ">", "^", "v". ამ სტრიქონის სიგრძე არ უნდა აღემატებოდეს 20 000-ს. შემდეგ თქვენმა პროგრამამ უნდა ნაკითხოს მთელი რიცხვთა  $r, c$  წყვილი ( $0 \leq r \leq H - 1, 0 \leq c \leq W - 1$ ) - რობოტის კოორდინატები ინსტრუქციების შესრულების შემდეგ. შევნიშნოთ, რომ ყოველი მოთხოვნის (ინსტრუქციათა ყოველი მიმდევრობის) შესრულების შემდეგ რობოტი ყოველთვის ბრუნდება უკან  $(0, 0)$  უჯრაში.
- როცა თქვენ უკვე იცით ყუთის ადგილმდებარეობა, გამოიტანეთ "!", რომელსაც უნდა მოსდევდეს მთელი რიცხვთა  $r_b, c_b$  წყვილი - იმ უჯრის კოორდინატები, რომელშიც საგანძურიანი ყუთია მოთავსებული ( $0 \leq r_b \leq H - 1, 0 \leq c_b \leq W - 1$ ). ამის შემდეგ თქვენი პროგრამა უნდა გამოვიდეს რაიმე შემდეგი მოთხოვნების გაკეთების გარეშე. ეს საბოლოო გამოსვლა თქვენი ქულების დაანგარიშების დროს მოთხოვნად არ ჩაითვლება.

აუცილებლად განზმინდეთ სტანდარტული გამოტანა მორიგი მოთხოვნის გაცემის შემდეგ. წინააღმდეგ შემთხვევაში თქვენმა პროგრამამ შეიძლება მიიღოს ვერდიქტი Time Limit Exceeded. პითონში `print()` ავტომატურად იშლება. C++-ში გამოიყენეთ `cout << endl;` თუ იყენებთ `printf`-ს, ისარგებლეთ `fflush(stdout)`-ით.

გრაფერი არაადაპტიურია, რაც ნიშნავს, რომ ყუთის მდებარეობა განსაზღვრულია ინტერაქციის დაწყებამდე.

## შეზღუდვები და შეფასება

- $1 \leq H, W \leq 50$ .
- ყუთი არასდროს იქნება  $(0, 0)$  უჯრაში. ეს ნიშნავს, რომ  $H + W \geq 3$ .
- ყოველი მოთხოვნა შეიძლება შეიცავდეს არაუმეტეს 20 000 ინსტრუქციას.
- თქვენ შეგიძლიათ გაგზავნოთ არაუმეტეს 2 500 მოთხოვნა (საბოლოო პასუხის ბეჭდვა მოთხოვნად არ ითვლება).

თქვენი ამოხსნა გარკვეული რაოდენობის ტესტებით შემოწმდება. თუ თქვენი ამოხსნა ჩავარდა *რომელიმე* ამ ტესტზე, (ანუ, თუ ვერდიქტი იქნება: Wrong Answer, Runtime Error, Time Limit Exceeded და ა.შ.), თქვენ მიიღებთ 0 ქულას.

თუ თქვენი პროგრამა წარმატებით პოულობს ყუთის ადგილმდებარეობას *ყველა* ტესტისათვის, მაშინ მიიღებთ ვერდიქტს Accepted და თქვენი ქულები ასე გამოითვლება:

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

სადაც  $Q$  არის ნებისმიერ ტესტში გამოყენებული მოთხოვნების მაქსიმალური რაოდენობა. საბოლოო პასუხის დაბეჭდვა მოთხოვნად არ ითვლება. ქულა დამრგვალებული იქნება უახლოეს მთელამდე.

კონკრეტულად, რომ მიიღოთ 100 ქულა, თქვენმა პროგრამამ უნდა ამოხსნას თითოეული ტესტი არაუმეტეს  $Q = 2$  მოთხოვნით. ქვემოთ მოცემულ ცხრილში ნაჩვენებია  $Q$ -ს ზოგიერთი მნიშვნელობა და შესაბამისი ქულები.

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

## ტესტირების ინსტრუმენტი

თქვენი ამოხსნის ტესტირება რომ გაადვილდეს, ჩვენ გთავაზობთ მარტივ ინსტრუმენტს, რომლის ჩამონერაც შეგიძლიათ. იხილეთ "attachments" Kattis ამოცანის გვერდის ბოლოში. ინსტრუმენტის გამოყენება სავალდებულო არაა და ასევე თქვენ შეგიძლიათ მისი შეცვლა. ყურადღება მიაქციეთ, რომ ოფიციალური გრაფერი Kattis-ზე ტესტირების ამ ინსტრუმენტისაგან განსხვავებულია.

გამოყენების მაგალითი ( $H = 4$ ,  $W = 5$  და დამალული ყუთი  $r = 2$ ,  $c = 3$  პოზიციაში):

Python-ისათვის, ვთქვათ, `solution.py` (ჩვეულებრივ ეშვება როგორც `pypy3 solution.py`):

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

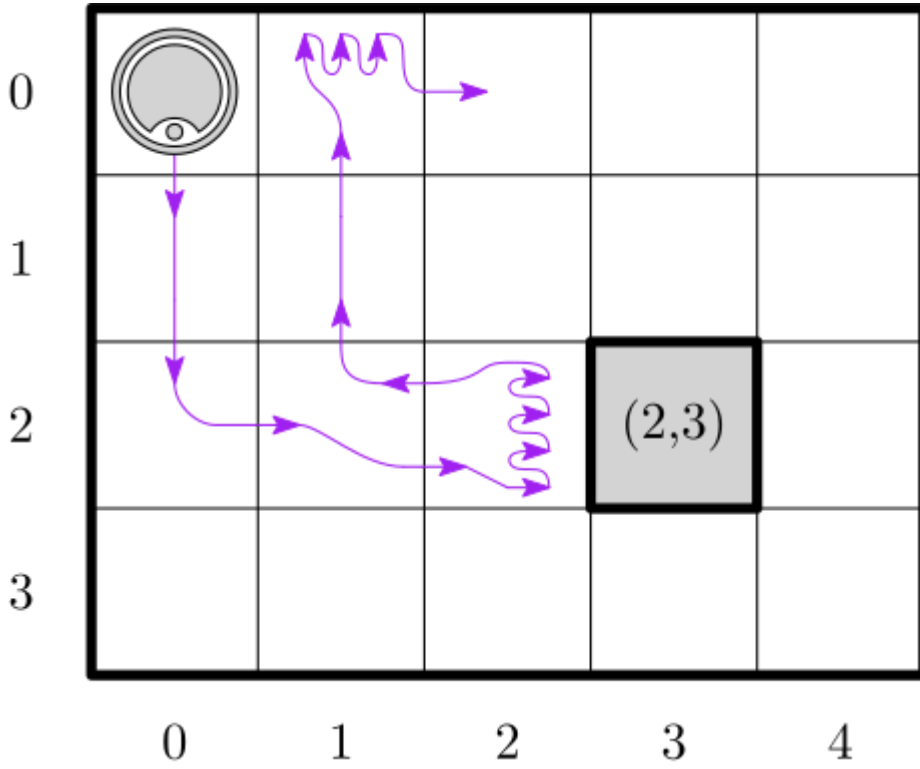
C++ პროგრამებისათვის თავიდან დააკომპილირეთ ისინი

(მაგალითად, გამოიყენეთ `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) და შემდეგ გაუშვით:

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

## მაგალითი

განვიხილოთ ტესტის მაგალითი. ბადეს აქვს სიმაღლე  $H = 4$ , სიგანე  $W = 5$  და ყუთი არის პოზიციაში  $(r, c) = (2, 3)$ . ქვემოთ მოცემულ ნახაზზე ნაჩვენებია რობოტის გზა, როცა ის ასრულებს პირველი მოთხოვნის ინსტრუქციებს: "`? vv>>>>>><^^^>`", რასაც მივყავართ იქამდე, რომ რობოტი აღმოჩნდება პოზიციაში  $(r, c) = (0, 2)$ . მეორე მოთხოვნის წინ რობოტი უნდა დაბრუნდეს უკან ზედა მარცხენა უჯრაში  $(0, 0)$ . შემდეგ ამოხსნა აგზავნის მეორე მოთხოვნას "`? >>>>>>>vvvvvvvvvv`", რომლის მიხედვითაც რობოტი აღმოჩნდება ქვედა მარჯვენა უჯრაში  $(r, c) = (3, 4)$ . ახლა ამოხსნა გადაწყვეტს გამოიცნოს სწორი პასუხი და წერს "`! 2 3`", რომელიც მართლაც ყუთის მდებარეობას წარმოადგენს.



გრაფერის გამოტანა	შენი გამოტანა
4 5	
	? \v>>>>>>><^^^>
0 2	
	? >>>>>>>>vwwwwwww
3 4	
	! 2 3