

## Find the Box

Problem Name	Find the Box
Time Limit	1 seconds
Memory Limit	1 gigabyte

Սա ինտերակտիվ խնդիր է:

Մայան ռոբոտաշինության ոլորտի հետազոտող է LTH-ում: Նա գիտի, որ համալսարանի նկուղային հարկում գանձ կա: Գանձը դրված է արկղի մեջ, որը գտնվում է մի սենյակում: Ցավոք, Մայան չի կարող ուղղակի գնալ ու վերցնել այդ արկղը: Այնտեղ շատ մուժ է, իսկ լույս տալով գնալը կասկածանք կհարուցի: Միակ ելքը նա տեսնում է արկղը հեռավար ղեկավարվող ռոբոտի միջոցով գտնելը:

Սենյակը կարելի է ներկայացնել  $H \times W$  չափի աղյուսակի մեջոցով, որտեղ տողերը համարակալված են 0-ից  $H - 1$  թվերով (վերևից ներքև), իսկ սյուները համարակալված են 0-ից  $W - 1$  թվերով (ձախից աջ), ինչը նշանակում է, որ ձախ վերևի վանդակը  $(0, 0)$ ն է, իսկ աջ ներքևի վանդակը  $(H - 1, W - 1)$ ն է: Արկղը գտնվում է վանդակներից մեկում (մնացած վանդակները դատարկ են): Ամեն գիշեր ռոբոտը սկսում է շարժվել ձախ վերևի վանդակից:

Ամեն գիշեր, Մայան ռոբոտին տալիս է հրահանգների տող, այն բաղկացած է 4 տեսակի սիմվոլներից՝ " $<$ ", " $>$ ", " $\wedge$ " և " $\vee$ ". Սլաքի ուղղությունը համապատասխանում է շարժման ուղղությանը՝ եթե ռոբոտը գտնվում է  $(r, c)$  վանդակում որի հարևան վանդակները ազատ են, " $<$ " տեղափոխում է դեպի ձախ՝  $(r, c - 1)$ , " $>$ " տեղափոխում է դեպի աջ՝  $(r, c + 1)$ , " $\wedge$ " տեղափոխում է դեպի վերև՝  $(r - 1, c)$ , և " $\vee$ " տեղափոխում է դեպի ներքև՝  $(r + 1, c)$ .

Պատերը և արկղը պինդ են: Երբ ռոբոտը փորձում է գնալ այնպիսի վանդակ, որը դուրս է սենյակի սահմաններից, կամ, որում արկղն է գտնվում, ապա ռոբոտը մնում է տեղում: Բոլոր հրահանգները կատարելուց հետո ռոբոտը Մայային կհայտնի իր տեղը և հետ կգնա ձախ վերևի վանդակ:

Մայան ուզում է արկղը գտնել կարճ ժամանակում:

## Փոխգործակցություն

Սա ինտերակտիվ խնդիր է:

- Ձեր ծրագիրը պետք է սկսի երկու ամբողջ թիվ կարդալով՝  $H$  և  $W$  (աղյուսակի չափերը): Արկղը չի գտնվի  $(0, 0)$  ում:
- Այնուհետև ձեր ծրագիրը պետք է սկսի հարցումներ տալ: Ամեն հարցում պետք է սկսվի “?” սիմվոլով, որին պետք է հետևի  $s$  ոչ դատարկ տողը, որը բաղկացած է “<”, “>”, “^”, “v” սիմվոլներից: Այդ տողի երկարությունը չպետք է գերազանցի 20 000ը: Հարցումը տալուց հետո Ձեր ծրագիրը պետք է կարդա երկու թիվ՝  $r, c$  ( $0 \leq r \leq H - 1, 0 \leq c \leq W - 1$ ) (ռոբոտի վերջնական կորդինատները): Նկատեք, որ ռոբոտը ամեն հարցումից հետո վերադառնում է  $(0, 0)$ :
- Երբ արկղի կորդինատները գտնված են, տպեք “!”, որին հետևում են երկու թվեր՝  $r_b, c_b$  (արկղի կորդինատները,  $0 \leq r_b \leq H - 1, 0 \leq c_b \leq W - 1$ ): Տպելուց հետո Ձեր ծրագիրը պետք է ավարտվի (այս գործողությունը չի մտնում հարցումների քանակի մեջ Ձեր միավորը հաշվելիս):

Հարցումները տպելուց հետո մի մոռացեք դատարկել արտածման բուֆերը, հակառակ դեպքում Ձեր ծրագիրը կարող է ստանալ Time Limit Exceeded: Python լեզվում `print()` ը մեխանիկորեն դատարկում է բուֆերը: C++ լեզվում, `cout << endl;` ը մաքրում է բուֆերը նոր տողի անցնելով; եթե `printf` եք օգտագործում, օգտագործեք `fflush(stdout)` հարմանը.

Գրեյդերը հարմարվող չէ, այսինքն արկղի կորդինատները ֆիքսված են Ձեր ծրագրի աշխատանքի սկզբից:

## Սահմանափակումներ և միավորներ

- $1 \leq H, W \leq 50$ :
- Արկղը չի գտնվում  $(0, 0)$  ում: Սա նշանակում է, որ  $H + W \geq 3$ :
- Ամեն հարցում պետք է բաղկացած լինի ամենաշատը 20 000 սիմվոլից:
- Դուք կարող եք օգտագործել ամենաշատը 2 500 հարցում:

Ձեր ծրագիրը կստուգվի մի քանի թեստի վրա: Եթե Ձեր լուծումը սխալվում է այդ թեստերից որևէ մեկում (օրինակ սխալ պատասխան (WA), սխալ կատարում ընթացքում (RTE), ժամանակի սահմանափակման գերազանցում (TLE), և այլն), ապա դուք կստանաք 0 միավոր:

Եթե Ձեր ծրագիրը ճիշտ է աշխատում բոլոր թեստերի համար, ապա դուք կստանաք AC, և Ձեր միավորը կհաշվի հետևյալ բանաձևով՝

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

որտեղ  $Q$  թիվը բոլոր թեստերում Ձեր կատարած մաքսիմալ հարցումների քանակն է: Պատասխանը տպելը դրա մեջ չի հաշվում: Միավորը կկլորացվի դեպի վերև:

Մասնավորապես, 100 միավոր ստանալու համար, Ձեր ծրագիրը պետք է օգտագործի ամենաշատը  $Q = 2$  հարցում: Հետևյալ աղյուսակը ցույց է տալիս  $Q$ -ի որոշ արժեքներ և համապատասխան միավորները:

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

## Թեստավորման գործիք

Ձեր լուրջ թեստավորումը հեշտացնելու համար, Ձեզ տրամադրվում է թեստավորման գործիք: Այն կարող էք գտնել “attachments” բաժնում: Այս գործիքի օգտագործումը պարտադիր չէ: Ձեր լուծման թեստավորման համար այս գործիքը չի օգտագործվելու:

Օրինակ թեստավորելու համար ( $H = 4$ ,  $W = 5$ , և արկղի կորդինատներն են  $r = 2$ ,  $c = 3$ ):

python լեզվի համար, եթե լուծուը `solution.py`ն է օգտագործվում է հետևյալ հրամանը`

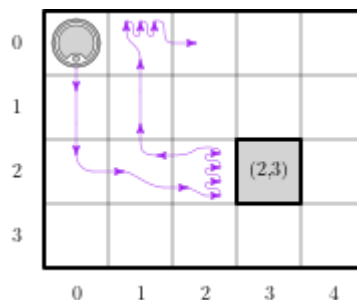
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

C++ լեզվի համար, պետք է սկզբից կոմպիլացնել (Օրինակ `g++ -std=gnu++17 solution.cpp -o solution.out`) հետո օգտագործել հետևյալ հրամանը`

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

## Օրինակ

Սենյակի չափերն են  $H = 4$  և  $W = 5$ , իսկ արկղը գտնվում է  $(r, c) = (2, 3)$  վանդակում: Հետևյալ նկարը ցույց է տալիս ռոբոտի շարժումը առաջին հրամանից` “? vv>>>>>>> <^^^^^” հետո, որի վերջում ռոբոտը գտնվում է  $(r, c) = (0, 2)$  վանդակում: Երկրորդ հարցումից առաջ ռոբոտը կվերադառնա  $(0, 0)$  վանդակ: Որից հետո տրվում է ևս մեկ հարցում` “? >>>>>>>vvvvvvvvvv”, որի վերջում ռոբոտը գտնվում է  $(r, c) = (3, 4)$  վանդակում: Դրանից հետո ծրագիրը գուշակություն է անում և տպվում է “! 2 3”, դրանք արկղի ճիշտ կորդինատներն են:



grader output	your output
4 5	
	? w>>>>>><^>>>>>>
0 2	
	? >>>>>>>vwwwwwww
3 4	
	! 2 3