

Leia kirst

Ülesande nimi	Find the Box
Ajalimiit	1 sekund
Mälulimiit	1 gigabait

Maj on Lundi Ülikoolis töötav robotikateadlane. Ta on kuulnud ülikooli keldris asuvast väärtuslikust varandusest. Varandus asub sügaval maa all tühjas ruumis kirstus. Paraku ei saa Maj lihtsalt kirstu otsima minna. Keldris on väga pime ning sinna valgusallikaga minemine näiks kahtlane. Tema ainus allesjäänud moodus varanduse leidmiseks on juhtida keldris asuvat robottolmuimejat.

Keldrit saab väljendada $H \times W$ ruudustikuna, mille read on nummerdatud 0-ist $H - 1$ -ni (ülevalt alla) ning mille veerud on nummerdatud 0-ist $W - 1$ -ni (vasakult paremale), seega ülemine vasakpoolne ruut on $(0, 0)$ ning alumine parempoolne ruut $(H - 1, W - 1)$. Varandusega kirst on mingis teadmata ruudus, mis ei ole $(0, 0)$. Igal ööl alustab robottolmuimeja oma tööd ülemises vasakpoolses ruudus ning liigub keldris ringi.

Igal ööl saab Maj anda robotile juhised ringi sõitmiseks sõnena, mis koosneb märkidest " $<$ ", " $>$ ", " \wedge " ja " \vee ". Täpsemalt, kui robot on ruudul (r, c) , mille ühelgi küljel pole takistust, siis liigutab " $<$ " roboti vasakule ruudule $(r, c - 1)$, " $>$ " liigutab roboti paremale ruudule $(r, c + 1)$, " \wedge " liigutab roboti üles ruudule $(r - 1, c)$ ning " \vee " liigutab roboti alla ruudule $(r + 1, c)$.

Keldri seinast läbi sõita ei saa, nii et kui robot üritab mõttelisest ruudustikust väljuda, ei juhtu mitte midagi. Ka kirst on läbimatu ning seda pole võimalik liigutada. Iga öö lõpus teatab robot enda asukoha ning naaseb ülemisse vasakusse nurka.

Aeg on raha, seega tahab Maj leida kirstu pärast võimalikult väikest arvu öid.

Suhtlus

See on interaktiivne ülesanne.

- Sinu programm peaks algul lugema rea, mis sisaldab kahte täisarvu H ja W , mis väljendavad ruudustiku kõrgust ja laiust.

- Seejärel peaks sinu programm suhtlema hindajaga. Iga päringu alguses peaks programm väljastama küsimärgi "?", millele järgneb mittetühi sõne s , mis koosneb märkidest "<", ">", "^" ja "v". Selle sõne pikkus võib olla kõige rohkem 20 000 märki. Seejärel peaks sinu programm lugema kaks täisarvu r ja c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), roboti asukoht (rida ja veerg) vahetult pärast juhiste järgimist. Pane tähele, et robot naaseb pärast iga päringut ruudule $(0, 0)$.
- Kui sinu programm teab kirstu asukohta, peaks see väljastama "!" ja seejärel täisarvud r_b, c_b , vastavalt kirstu asukohta kirjeldavad rida ja veerg ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). Seejärel peaks programm edasise suhtluseta töö lõpetama. Punktisumma määramisel ei loeta sellist viimast rida päringute hulka.

Veenduge, et programm tühjendaks väljundpuhvri peale iga päringu esitamist, vastasel juhul võib programm saada hinnangu "Ajalimiit ületatud" (*Time Limit Exceeded*). Pythonit kasutades tühjendab `print()` väljundpuhvri automaatselt. C++ kasutades väljastab `cout << endl;` reavahetuse ning tühjendab seejärel väljundpuhvri; kui kasutad `printf`, kasuta `fflush(stdout)`.

Kirstu asukoht on enne suhtluse algust fikseeritud (suhtlus ei muuda kirstu asukohta).

Piirangud ja hindamine

- $1 \leq H, W \leq 50$
- Kirst pole kunagi ruudul $(0, 0)$. See tähendab, et $H + W \geq 3$.
- Iga juhiste jada (sõne) võib koosneda kõige rohkem 20 000 juhisest.
- Sinu programm võib esitada kuni 2 500 juhiste jada (vastuse väljastamist ei loeta nende hulka).

Sinu lahendust testitakse hulga testidega. Kui sinu lahendus ei läbi *ükskõik missugust* neist testidest (näiteks väljastades vale kirstu asukoha (*Wrong Answer*), kokku joostes (*Runtime Error*), ületades ajalimiiti (*Time Limit Exceeded*) vms), saab lahendus 0 punkti ning vastava hinnangu.

Kui sinu programm leiab edukalt kirstu asukoha *kõigis* testides, saab see esitus hinnangu AC (korras) ning punktisumma, mis arvutatakse järgnevalt:

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

kus Q on maksimaalne esitatud päringute arv üle kõigi testide. Vastuse väljastamist ei loeta päringute hulka. Punktisumma ümardatakse lähima täisarvuni.

Seejuures peab programm 100 punkti teenimiseks lahendama iga testi, kasutades kõige enam $Q = 2$ päringut. Järgnev tabel kirjeldab mõnd Q väärtust ning sellele vastavat punktisummat.

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

Testimistööriist

Lahenduse testimiseks on sulle antud allalaadimiseks lihtne tööriist. Vaata manuste ("*attachments*") osa Kattise lehe alaosas. Selle tööriista kasutamine on valikuline ning seda on lubatud muuta. Pange tähele, et ametlik hindamisprogramm Kattises erineb antud testimistööriistast.

Näidiskasutus (kus $H = 4$, $W = 5$, ning peidetud kirst on asukohas $r = 2$, $c = 3$):

Pythoni programmi, näiteks `lahendus.py` puhul (niisama käivitamiseks `python3 lahendus.py`):

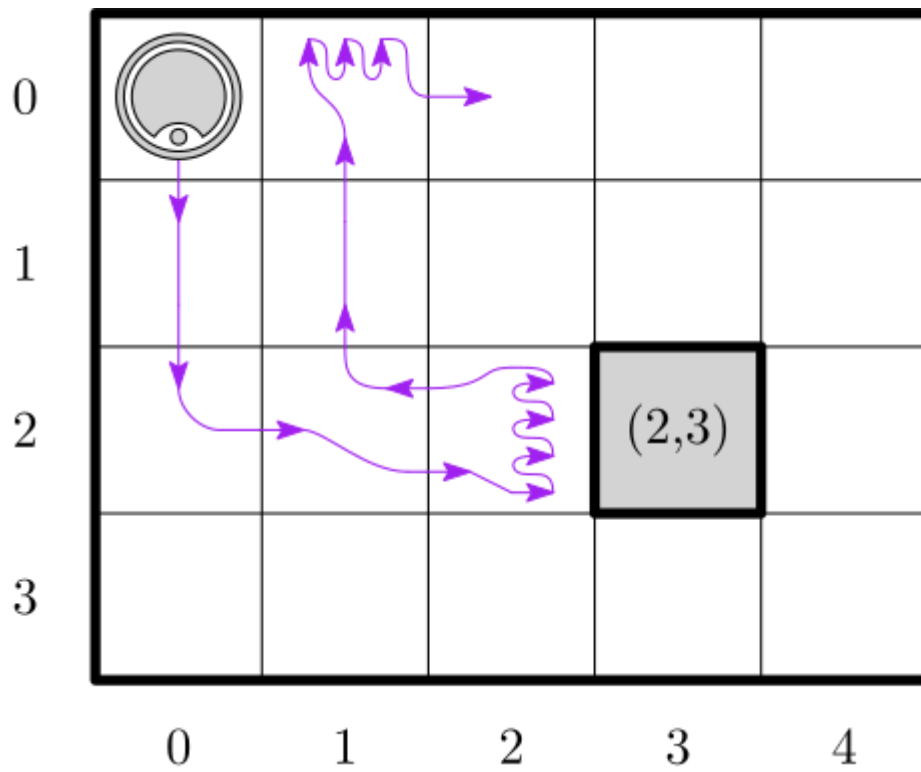
```
python3 testing_tool.py python3 lahendus.py <<<"4 5 2 3"
```

C++ programmi puhul esmalt kompileeri oma programm (näiteks käivitades `g++ -std=gnu++17 lahendus.cpp -o lahendus.out`) ning seejärel käivita:

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

Näidis

Olgu näiteks ruudustiku kõrgus $H = 4$ ja laius $W = 5$ ning kast asukohas $(r, c) = (2, 3)$. Allpool olev joonis näitab roboti teekonda, kui ta järgib juhiseid esimesest päringust "`? vv>>>>>><^^^^^>`", mille tulemusena jõuab robot asukohta $(r, c) = (0, 2)$. Enne teist päringut naaseb robot taas üles vasakule nurka $(0, 0)$. Seejärel annab näidislahendus päringu "`? >>>>>>>vvvvvvvvvvv`", mille järel robot jõuab alumisse parempoolsesse nurka $(r, c) = (3, 4)$. Seejärel on näidises pakutud vastust, kirjutades "`! 2 3`", mis on kasti õige asukoht.



hindaja väljund	näidisväljund
4 5	
	? w>>>>>>><^>>>>>>
0 2	
	? >>>>>>>v>>>>>>>
3 4	
	! 2 3