

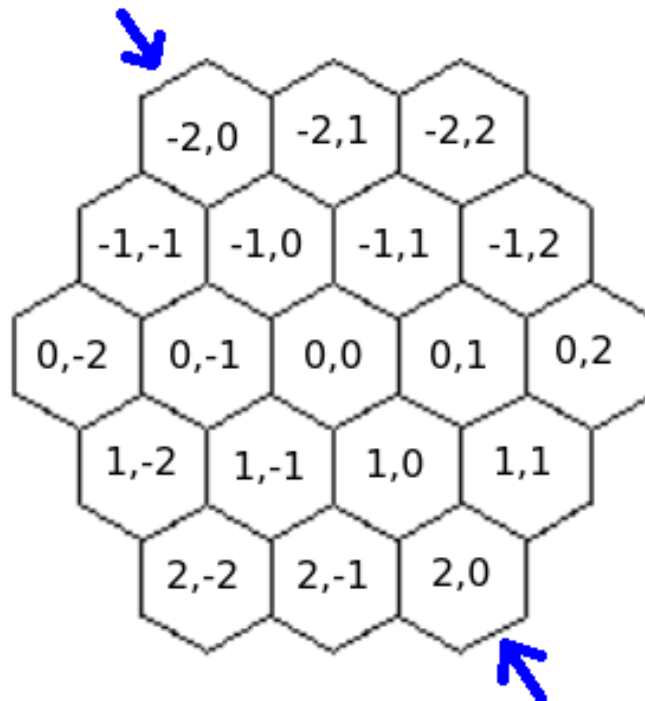
Ավազակ լեմուրը

Ժամանակի սահմանափակում՝ 0.3 վայրկյան
Հիշողության սահմանափակում՝ 256 MB
Կշիռը՝ 100 միավոր

Շարադրանք

Ավազակ լեմուրն ազատության մեջ է !!!

Ավազակ լեմուրը կենդանաբանական այգիից փախուստի է դիմել և հիմա գտնվում է ազատության մեջ: Նա գտնվում է կանոնավոր վեցանկյան տեսք ունեցող վանդակավոր դաշտում, որի ամեն կողը բաղկացած է 20 վեցանկյուններից: Ավելի փոքր դաշտի օրինակը տես ստորև (3 վանդակ երկարությամբ կողով): Լեմուրը սկզբում գտնվում է 0, 0 կորդինատներով վանդակում: Ամեն քայլում, դուք կարող եք թակարդ տեղադրել յուրաքանչյուր ազատ վանդակի ներսում, բացի լեմուրի ներկայիս վանդակից: Լեմուրը մեկ քայլի ընթացքում կարող է տեղափոխվել իր ներկայիս վանդակի հարևան վանդակներից յուրաքանչյուրը, որտեղ թակարդ չկա: Ձեր խնդիրն է որսալ լեմուրին՝ նրան գցելով թակարդներից բաղկացած բազմանկյան (փակ բեկյալի) մեջ: Լեմուրը հաղթում է, եթե կարողանում է հասնել վեցանկյան եզրային վանդակներից ինչ-որ մեկին: Քայլերը կատարվում են հերթով. սկսում է լեմուրը: Դուք ունեք M թակարդ: Լեմուրը շատ ուժ է ծախսել կենդանաբանական այգուց փախչելու վրա և հիմա չի կարող օպտիմալ քայլել: Լեմուրը իր ամեն քայլին գտնում է դեպի դուրս տանող կարճագույն ճանապարհներից որևէ մեկը (պատահականորեն) և քայլ անում այդ ուղղությամբ:



Պահանջվում է իրականացնել

```
pair<int, int> step(pair<int, int> cat) { ... } $ունկցիան, որը որպես արգումենտ ստանում է կատվի հերթական դիրքը և վերադարձնում ձեր կողմից փակվող վանդակի կորդինատները: Դուք կարող եք իրականացնել void init(int m) { ... } $ունկցիան, որը կկանչվի մեկ անգամ՝
```

ամենասկզբում: Նրան կփոխանցվի ձեզ հասանելի թակարդների քանակը:
Ընթացիկ որոշ փոփոխականներ/ինֆորմացիա պահելու համար կարող եք
օգտագործել գլոբալ փոփոխականների տիրույթը: DOWNLOAD grader.h
(/static/lemur/grader.h)

YOUR CODE

```
#include "grader.h"

...

void init(int m) {
    ...
}

pair<int, int> step(pair<int, int> cat) {
    ...
}
```

Ենթախնդիրներ

- Ենթախնդիր 1 (**45 միավոր**) $m = 200$
- Ենթախնդիր 2 (**55 միավոր**) $m = 20$