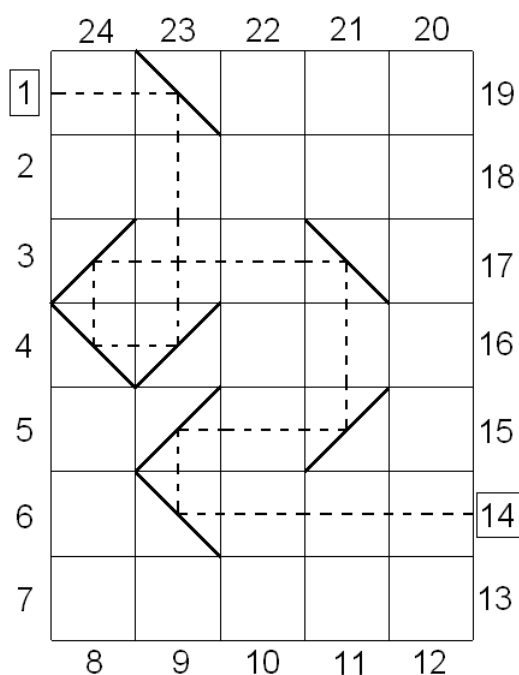
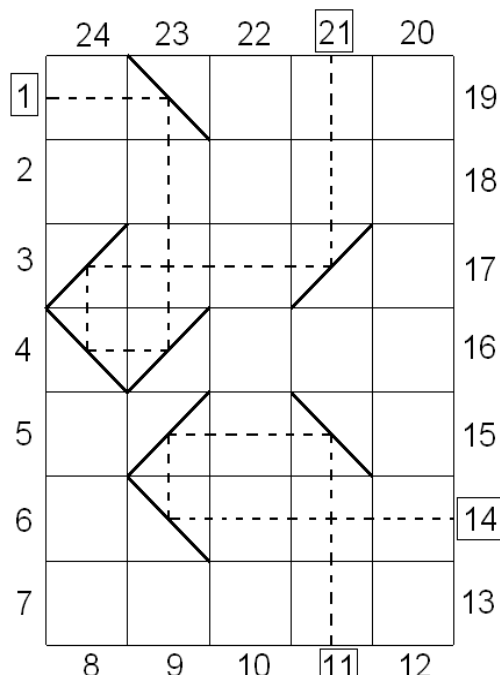


Задача А3. Огледален лабиринт

Хензел и Гретел са затворени от злата вещица в нейния огледален лабиринт. Лабиринта представлява правоъгълна стая с размери **NxM**. Стаята е разделена на квадрати с ширина и дължина единица. Във всеки такъв квадрат се намира по едно огледало, което може да е поставено в едно от три състояния – да е скрито, погледнато отгоре да се намира по левия диагонал или по десния диагонал на квадрата. По ръбовете на тази стая и извън нея се намират клетки, които са номерирани както е показано на фигурите. Хензел и Гретел са затворени в две такива клетки. От своя затвор те могат да виждат само през малък проце́п в клетката. Номерата на всяка клетка са написани точно на височината на проце́па. Злата вещица е обещала на Хензел и Гретел, че ще ги освободи ако успеят да ѝ решат една задача (е тя всъщност няма такова намерение понеже все пак е зла). Задачата е следната : първо вещицата наглася огледалата така че Хензел и Гретел да се виждат един друг, след това тя сменя състоянията на някои огледала (не повече от **6**) така че Хензел и Гретел да не се виждат повече и накрая дава на Хензел списък с **P** огледала, в който се намират и сменените огледала. Списъкът не указва кое огледало на кое квадратче от лабиринта отговаря и в него огледалата са номерирани с числата от **0** до **P-1**. Хензел трябва да намери такава комбинация от до шест огледала, че като ги постави в определени състояния (които пак той трябва да определи) двамата с Гретел отново да се виждат. Единствената информация, с която разполага Хензел след всяко налущване са номерата на клетките, които виждат той и Гретел при съответната промяна на огледалата. Ако налущването е неуспешно огледалата се връщат както са били преди промяната и следва ново налущване. Вашата задача е да помогнете на Хензел и Гретел като напишете програма **mirrors.exe**, която въвежда чрез външен модул числата **N**, **M** и **P**, и пак използвайки модула налущва комбинация от огледала и техните състояния, които решават задачата.



Фигура 1



Фигура 2

На примерния тест Хензел се намира в клетка номер 1, а Гретел в клетка номер 14. На фигура 1 е показано състоянието на лабиринта преди вещицата да премести някои огледала така че Хензел и Гретел да не се виждат. На фигура 2 е показано коя клетка вижда Хензел след промяната (това е клетка с номер 21), и коя клетка вижда Гретел (клетка с номер 11).

След стартирането си програмата ви трябва да извика **само веднъж** подпрограмата **init()**, която инициализира библиотеката, след това получавате **N, M** и **P** чрез функциите **getN()**, **getM()** и **getP()**. Когато сте готови с някакво налущване викайте подпрограмата **getView(...)** като за първи параметър ѝ давате броя огледала (между **0** и **6**) чиито състояния искате да определите, за втори параметър масив с поне толкова елементи колкото са огледалата – в този масив трябва да запишете номерата на огледалата (номерата са между **0** и **P-1** включително), третия параметър е отново масив от цели числа съдържащ за всяко огледало състоянието в което искате да го поставите (0-легнало; 1-наклонено надясно; 2-наклонено наляво), четвъртия параметър е променливата, в която се записва номера на клетката която вижда Хензел, петия параметър е променливата в която се записва номера на клетката която вижда Гретел. Ако някой от параметрите е извън допустимите стойности модула прекратява изпълнението на програмата със съобщение за грешка и получавате 0 точки за този тест. Програмата ви приключва успешно когато намерите подходящи огледала, като подпрограмата **getView(...)** се грижи за терминиране на Вашата програма.

Ограничения : $0 < N, M, P \leq 100$.

ЗА ПИШЕЩИТЕ НА FreePascal

Библиотека: (module.ppu, module.o)

```

procedure init;
function getN : integer;
function getM : integer;
function getP : integer;
procedure getView(num:integer;const mirrors, states: array of integer;
                var henzel_view, gretel_view : integer);

```

Програмата **example.pas** е пример за използване на тази библиотека.

Инструкции: За да компилирате вашия файл `mirrors.pas`, включете оператора `uses module`; в изходния текст и компилирайте с `fpc -So -O2 -XS mirrors.pas`

ЗА ПИШЕЩИТЕ НА GNU C/C++

Библиотека: (`module.h`, `module.o`)

```

void init();
int getN();
int getM();
int getP();
void getView(int num, int *mirrors, int *states,
            int *henzel_view, int *gretel_view);

```

Програмата `example.c` е пример за използване на тази библиотека.

Инструкции: За да компилирате вашия файл `mirrors.c` или `mirrors.cpp`, използвайте

`#include "module.h"` в изходния текст и компилирайте с:

```

gcc -O2 -static mirrors.c module.o -lm -o mirrors.exe
gXX -O2 -static mirrors.cpp module.o -lm -o mirrors.exe

```

За пишещите на C/C++ в RHIDE: Непременно задайте на Options->Linker configuration стойност `module.o`.

Инструкции за тестване :

Модула чете входните данни от файла **mirrors.in**. На първия ред на този файл са записани числата *N* и *M*. На следващите *N* реда има по *M* числа разделени с интервали като всяко едно от тези числа може да приема една от три стойности : 0 – скрито огледало; 1 – огледало наклонено надясно; 2 – огледало наклонено наляво. Следва ред с две числа – номерата на клетките, в които се намират Хензел и Гретел. На следващия ред е числото *P*, следвано от *P* реда описващи координатите на реда и на стълба на всяко от огледалата, които могат да се променят от Хензел (горния ляв ъгъл на лабиринта е с координати (0, 0)). Номерата на огледалата са според това кога се появяват във входния файл (първото е с номер 0, второто с номер 1 и т.н).

Ето и как изглежда **mirrors.in** за примера от фигурите :

```

7 5
0 2 0 0 0
0 0 0 0 0
1 0 0 1 0
2 1 0 0 0
0 1 0 2 0
0 2 0 0 0
0 0 0 0 0
1 14
4
2 3
3 1
4 1
4 3

```

Заб. 1 Освен двете огледала, които вещицата е променила могат да се променят още и огледалата с координати (3, 1) и (4, 1).

Заб. 2. Резултатът от работата на програмата се извежда на стандартния изход. "Correct Answer!" означава, че програмата ви е открила вярна комбинация от огледала.