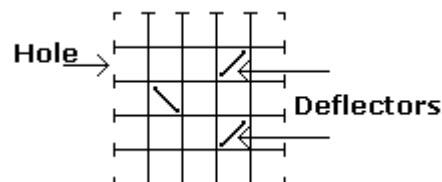




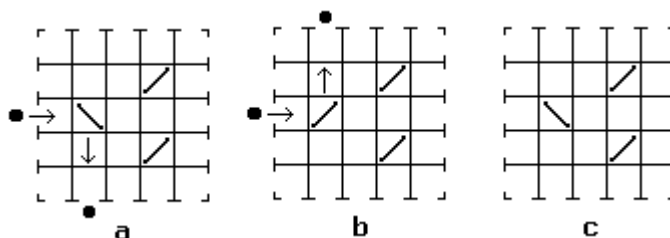
ИГРА С ЧЕРНА КУТИЯ

Черна кутия с квадратна форма е поставена върху маса, легнала на една от големите си стени. На всяка от четирите малки стени има по n отвора (общо $4n$ отвора). Във всеки отвор може да бъде хвърлена топка. Хвърлената топка, рано или късно излиза от кутията през някой от отворите, възможно и през този, в който е хвърлена.

Вътрешността на кутията можем да си представим като квадратна решетка с размери n на n . Отворите са върху клетките, с които започва и завършва всеки ред и стълб на решетката. Всяка от клетките на решетката е или празна или в нея има *отражател*. Отражателят е устройство, поставено по диагонала на клетката, което променя посоката на движение на падаща върху него топка на 90° . Вижте примера вдясно, за кутия с размери 5×5 .



Когато хвърлим топка в кутията, тя продължава да се движи в посоката, в която е била хвърлена, докато се удари в отражател или излезе от кутията. Когато топката се удари в отражател, тя променя посоката си на движение, а отражателят сменя положението си на 90° . Примерът по-долу показва действието на отражателя.



- 0 Топка е хвърлена в отвор; топката среща отражател и променя посоката на движение.
- 1 След удара на първата топка отражателят е сменил положението си. Следваща топка, която се удари в този отражател, ще се отрази в посока противоположна на тази, в която се е отразила първата топка.
- 2 Отражателят променя положението си след всеки удар.

Когато отражател е ударен от топка, той издава звук. Броят на отразяванията на топката може да се определи, като се брой колко пъти се е чул звукът от удар. Може да се докаже, че топката винаги ще напусне кутията. Кутията има бутон, натискането на който възстановява началните положения на всички отражатели.

ЗАДАЧА

Ще получите интерфейс към 15 черни кутии, под формата на библиотеки от функции на Pascal и C/C++. Трябва да определите вътрешността на всяка от кутиите колкото може по-точно и да изпратите резултата, записан във файл. Ще получите и възможност да зададете своя собствена кутия за тестване.

ОГРАНИЧЕНИЯ

$$1 \leq n \leq 30$$



ИЗХОД

Трябва да предадете файл съдържащ следните данни за всяка от 15-те черни кутии:

blackboxK.out	ОПИСАНИЕ
<pre>#FILE blackbox K/. .\... .../. .?.?.</pre>	РЕД 1: Заглавният ред на файла трябва да е #FILE blackbox K където K (от 1 до 15) е номерът на кутията, която изследвате. n РЕДА: Всеки от тях описва един ред на кутията, започвайки от най-горния до най-долния и трябва да съдържа точно <i>n</i> знака. Всеки знак съответства на един стълб (взети отляво надясно). • '.' означава, че клетката е празна. • '/' означава, че клетката съдържа отражател в начално положение '/' • '\' означава, че клетката съдържа отражател в начално положение '\' • '?' означава, че не сте могли да определите първоначалното съдържание на тази клетка.

БИБЛИОТЕКА

Дадена е библиотека, съдържаща следните функции:

ФУНКЦИЯ	ОПИСАНИЕ
PASCAL <pre>function Initialize(box: integer): integer;</pre> C/C++ <pre>int Initialize(int box);</pre>	Инициализира библиотеката. Трябва да се извика един път в началото на програмата. Връща броя отвори от всяка страна на кутията. Параметърът <i>box</i> трябва да съдържа цяло число между 1 и 15, означаващо кутията, която искате да изследвате, или 0, ако искате да изследвате кутия, която сами сте създали.
PASCAL <pre>function throwBall(holeIn, sideIn: integer; var holeOut, sideOut: integer): longint;</pre> C <pre>int throwBall(int holeIn, int sideIn, int *holeOut, int *sideOut);</pre> C++ <pre>int throwBall(int holeIn, int sideIn, int &holeOut, int &sideOut);</pre>	Хвърля топка в кутията през отвор номер <i>holeIn</i> от страната <i>sideIn</i>, като страните са номерирани по следния начин: 1 – горна стена, 2 – дясна стена, 3 – долна стена и 4 – лява стена. За всяка страна отворите са номерирани (отляво надясно или отгоре надолу) започвайки от номер 1. В <i>holeOut</i> и <i>sideOut</i> ще получите номера на страната и номера на отвора, от който топката излиза от кутията. Функцията <i>throwBall</i> връща броя на отражателите, ударени от топката.
PASCAL <pre>procedure ResetBox;</pre> C/C++ <pre>void ResetBox();</pre>	Възстановява началното положение на всеки отражател в изследваната кутия.
PASCAL <pre>procedure ToggleDeflectors;</pre> C/C++ <pre>void ToggleDeflectors();</pre>	Променя ориентацията на всеки отражател в кутията.
PASCAL <pre>procedure Finalize;</pre> C/C++ <pre>void Finalize();</pre>	Приключва нормално диалога с кутията. Трябва да бъде извикана в края на програмата.

За да можете да използвате библиотеката в своята програма, направете следното:

- **FreePascal:** В директорията на задачата ще намерите файловете `pbbplib.o` и `pbbplib.ppu`. За да ги използвате, включете в кода: `uses pbbplib;`
Файлът `pblackbox.pas` е пример за това, как да използвате библиотеката.



- **C:** В директорията на задачата ще намерите файловете `cbblib.o` и `cbblib.h`. За да ги използвате, включете в кода: `#include "cbblib.h"`
 Файлът `cblackbox.c` е пример за това, как да използвате библиотеката. За да компилирате своята програма трябва да използвате следната команда:
`gcc -o yourprogram cbblib.o yourprogram.c`
- **C++:** В директорията на задачата ще намерите файловете `cppbblib.o` and `cppbblib.h`. За да ги използвате, включете в кода: `#include "cppbblib.h"`
 Файлът `cppblackbox.cpp` е пример за това, как да използвате библиотеката. За да компилирате своята програма трябва да използвате следната команда:
`g++ -o yourprogram cppbblib.o yourprogram.cpp`

ЗАБЕЛЕЖКА: Във всеки момент само една работеща програма може да използва библиотеката.

ПРИМЕРЕН ДИАЛОГ

Един примерен диалог за кутията от предната фигура може да бъде:

ИЗВИКАНА ФУНКЦИЯ	СТОЙНОСТ ВЪРНАТА ОТ ФУНКЦИЯТА
<code>Initialize(0);</code>	Приемаме, че кутията, която използваме е точно тази във фигурата. Функцията връща 5, означаващо че $n = 5$
PASCAL <code>throwBall(3,4,holeOut,sideOut);</code> C <code>throwBall(3,4,&holeOut,&sideOut);</code> C++ <code>throwBall(3,4,holeOut,sideOut);</code>	Хвърлена е топка в отвор номер 3 (третият отгоре надолу) на лявата страна. Връща 1, означаващо че топката е ударила един отражател. Когато функцията върне своя отговор, <i>holeOut</i> ще има стойност 2, а <i>sideOut</i> ще е равно на 3, което означава, че топката е излязла през отвор номер 2 (вторият отляво надясно) на долната страна на кутията.

ЕКСПЕРИМЕНТИРАНЕ

Ако подадете стойност 0 на функцията `Initialize`, библиотеката ще прочете съдържанието на кутията от файла `blackbox.in`. По този начин може да експериментирате с библиотеката. Форматът на файла `blackbox.in` е описан по-долу.

<code>blackbox.in</code>	ОПИСАНИЕ
5	РЕД 1: Съдържа n – броя на отворите от всяка страна.
3	РЕД 2: Съдържа цяло число d – броя на отражателите в кутията.
2 3 \	СЛЕДВАЩИ РЕДОВЕ: Трябва да има по един ред за всеки отражател в кутията. Всеки ред съдържа две разделени с интервал цели числа – номера на стълба и реда на отражателя, както и един знак, разделен с интервал и записан след второто число, който може да бъде <code>'/'</code> или <code>'\'</code> и означава началната позиция на отражателя.
4 2 /	
4 4 /	

ЗАБЕЛЕЖКА: Съдържанието на файла в горния пример описва черната кутия от горната фигура на стр. 1.



СЪОБЩЕНИЯ ЗА ГРЕШКИ

В случай на неправилно използване библиотеката ще изведе съобщение за грешка на стандартния изход за грешки. Възможните грешки са посочени в следната таблица.

СЪОБЩЕНИЕ	ОЗНАЧАВА
ERR 1 More than one app	Само една програма може да използва библиотеката в даден момент.
ERR 2 Invalid box	Номерът на кутията не е число от 0 до 15.
ERR 3 Invalid deflector	Файлът blackbox.in съдържа недопустими координати на отражател
ERR 4 Invalid symbol	Файлът blackbox.in съдържа недопустим знак
ERR 5 Invalid size	Недопустим размер на кутия.
ERR 6 Invalid input hole	Невалиден номер на страна или невалиден номер на отвор.
ERR 7 ALARM	Моля, извикайте техническия персонал!

ОЦЕНЯВАНЕ

За всяка кутия, трябва да изпратите текстов файл, който описва вътрешността ѝ колкото може по-добре. За всяка кутия:

- ако изпратеният файл съдържа знак '.', '/' или '\' на неправилно място, получавате нула точки за тази кутия
- Нека B_m е максималният брой позиции, чието начално съдържание е правилно намерено в решението на някой състезател, и нека B_y е броят позиции, чието начално съдържание сте намерили правилно. Тогава Вашият резултат за тази кутия ще бъде:

$$100 B_y / B_m$$

ЗАБЕЛЕЖКА: Съществува решение за тази задача, което може да открие 100% от началното съдържание на всяка от кутиите, за време по-малко от 8 минути, без човешка намеса.