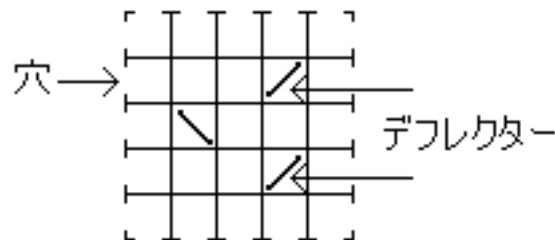




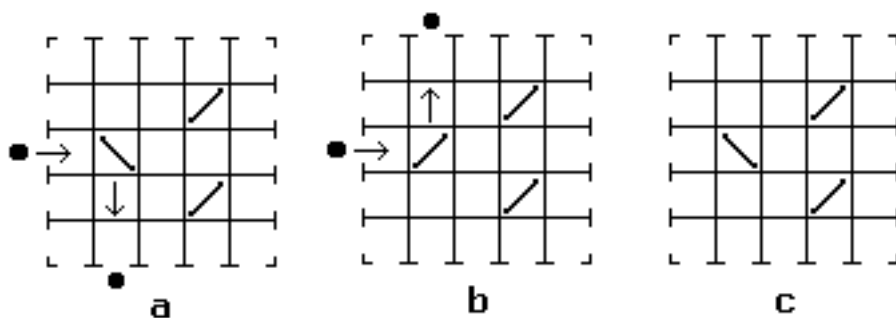
## ブラックボックスゲーム (A BLACK BOX GAME)

ブラックボックスゲームは、テーブルの上に置いた正方形の黒い箱を使って行う。正方形の四辺には各  $n$  個 (計  $4n$  個) の穴があいており、そこからボールを投げ入れることができるようになっている。箱に入れたボールはじきに  $4n$  個のどれかの穴から出てくる。入れたのと同じ穴から出てくる場合もある。

黒い箱の中は  $n \times n$  のグリッドとみなすことができる。穴は各行と各列の両端に対応する。箱の各マスは空であるか、デフレクターが置かれている。デフレクターはボールの向きを  $90$  度変える機械である。例として図の  $5 \times 5$  の箱を考える。



箱に入ったボールはデフレクターに当たるか箱の外に出るまで直線状に進む。ボールがデフレクターに当たると、ボールの方向が変わり、さらにデフレクターがトグルする (「トグルする」とは、方向が  $90$  度変わることをいう)。下の例はデフレクターの動作を示している。



- ボールを穴から投げ入れると、ボールはデフレクターに当たって進む向きが変わる。
- 最初のボールを投げた後、デフレクターがトグルした。新しいボールを同じ穴から投げ入れると、同じデフレクターに当たり、最初のボールとは反対の向きに反射する。
- デフレクターはボールが当たる度にトグルする。

デフレクターにボールが当たると音が鳴る。この音の回数を数えることで、ボールが反射した回数を知ることができる。ボールは必ず箱の外に出てくることが証明できる。箱には状態を初期状態に戻すボタンとすべてのデフレクターをトグルするボタンがついている。

## 課題

15 個の黒い箱に対するインターフェースが Pascal と C/C++ の関数ライブラリの形で提供される。これらの各箱の内部を可能な限り決定し、各々を記述したファイルを提出せよ。また、テストのために自分で黒い箱を定義する方法も提供される。

## 制約

$$1 \leq n \leq 30$$

## 出力

15 個の黒い箱の各々について、次のデータを書いたファイルを提出せよ。

blackboxK.out	説明
<pre>#FILE blackbox K ..... .../. .\... .../. .?.?.</pre>	<p>1 行目にはファイルヘッダを書く。ファイルヘッダは、<math>K</math> (1 以上 15 以下) を箱の番号として</p> <pre>#FILE blackbox K</pre> <p>という形をしていなければならない。</p> <p>次の <math>n</math> 行の各々は箱の行を上から下の順に表す。各行はちょうど <math>n</math> 個の文字を含まなければならない、各文字が左から右の順に各列に対応する。</p> <ul style="list-style-type: none"><li>「<code>.</code>」はそのマスが空であることを表す。</li><li>「<code>/</code>」はそのマスにデフレクターがあり、初期状態で「<code>/</code>」の方向を向いていることを表す。</li><li>「<code>\</code>」はそのマスにデフレクターがあり、初期状態で「<code>\</code>」の方向を向いていることを表す。</li><li>「<code>?</code>」はあなたがそのマスの初期状態を決定できなかったことを表す。</li></ul>

# ライブラリ

関数	説明
PASCAL function Initialize(box: integer): integer; C/C++ int Initialize(int box);	ライブラリを初期化する。プログラムの開始時に 1 度呼ばなければならない。戻り値は箱の各辺の穴の個数 $n$ を表す。 パラメータ box には、使いたい箱を表す 1 から 15 までの整数が、自分で作った箱を使いたい場合は 0 を指定しなければならない。
PASCAL function throwBall(holeIn, sideIn: integer; var holeOut, sideOut: integer): longint; C int throwBall(int holeIn, int sideIn, int *holeOut, int *sideOut); C++ int throwBall(int holeIn, int sideIn, int &holeOut, int &sideOut);	箱の辺 sideIn の holeIn 番目の穴からボールを投げ入れる。辺は 1:上, 2:右, 3:下, 4:左 の番号で表す。穴は辺ごとに左から右または上から下に 1 から順に番号が付いている。ボールが箱から出てくる場所の穴と辺の番号が holeOut と sideOut に返される。関数 throwBall の戻り値は、ボールがデфлекターに当たって音が鳴った回数である。
PASCAL procedure ResetBox; C/C++ void ResetBox();	箱の中のすべてのデфлекターを初期状態に戻す。
PASCAL procedure ToggleDeflectors; C/C++ void ToggleDeflectors();	箱の中のすべてのデфлекターをトグルする。
PASCAL procedure Finalize; C/C++ void Finalize();	箱とのやりとりを安全に (gracefully) 終了する。プログラムの終了時にこの関数を呼ぶべきである。

ライブラリを使うためには次のようにせよ。

FreePascal タスクディレクトリにファイル pbplib.o と pbplib.ppu があり、次の文を入れると使える。

```
uses pbplib;
```

ライブラリの使用例がファイル pblackbox.pas にある。

C 言語 タスクディレクトリにファイル cbplib.o と cbplib.h があり、次の文を入れると使える。

```
#include "cbplib.h"
```

ライブラリの使用例がファイル cblackbox.c にある。コードをコンパイルするには、

```
gcc -o yourprogram cbplib.o yourprogram.c
```

というコマンドを使う必要がある。

C++ タスクディレクトリにファイル `cppbplib.o` と `cppbplib.h` があり、次の文を入れると使える。

```
#include "cppbplib.h"
```

ライブラリの使用例がファイル `cppblackbox.cpp` にある。コードをコンパイルするには、

```
g++ -o yourprogram cppbplib.o yourprogram.cpp
```

というコマンドを使う必要がある。

注意 ライブラリを使うプログラムは、同時に 1 個しか実行できない。

## やりとりの例

次は前の図の箱に対するやりとりの例である。

関数呼び出し	関数が返す値
<code>Initialize(0);</code>	前の図の箱が使われるとすると、返り値は 5 で、 $n = 5$ であることを表す。
PASCAL <code>throwBall(3,4,holeOut,sideOut);</code>	ボールを左の辺の穴 3 (上から 3 番目の穴) から投げ入れる。返り値は 1 で、ボールが 1 個のデフレクターに当たったことを表す。関数から戻ったとき、 <code>holeOut</code> は 2、 <code>sideOut</code> は 3 になっており、ボールが下の辺の穴 2 (左から 2 番目の穴) から出てきたことを表す。
C <code>throwBall(3,4,&amp;holeOut,&amp;sideOut);</code>	
C++ <code>throwBall(3,4,holeOut,sideOut);</code>	

## 実験

関数 `Initialize` に整数 0 を渡すと、ライブラリは箱の内部の情報をファイル `blackbox.in` から読み込む。これによりライブラリで実験をすることができる。ファイル `blackbox.in` の形式を次に示す。

<code>blackbox.in</code>	説明
5	1 行目には各辺の穴の個数 $n$ を書く。
3	2 行目には箱の中のデフレクターの個数を表す整数 $d$ を書く。
2 3 \ 4 2 / 4 4 /	次の $d$ 行には、箱の中の各デフレクターに対応した行がなければならない。各行には 2 個の整数を空白で区切って書き、それぞれデフレクターの列と行を表す。2 番目の整数の後、空白を挟んで「/」か「\ \'\' の文字を書き、デフレクターの初期状態での方向を表す。

注意 この例の `blackbox.in` は 1 ページの冒頭にある箱を記述している。

## エラーメッセージ

例外的な場合には、ライブラリは標準エラー出力にエラーメッセージを出力する。起きうるエラーメッセージとその意味は次の通りである。

エラーメッセージ	意味
ERR 1 More than one app	一度に箱とやりとりをできるアプリケーションは 1 個だけである。すべてのアプリケーションを再実行し、一度に一つしか動かないようにしてください。
ERR 2 Invalid box	指定された箱の番号が 0 から 15 までの範囲に入っていない。
ERR 3 Invalid deflector	ファイル <code>blackbox.in</code> でデフレクターが無効な位置に置かれている。
ERR 4 Invalid symbol	ファイル <code>blackbox.in</code> に無効な記号が書かれている。
ERR 5 Invalid size	ファイル <code>blackbox.in</code> に書かれている箱の大きさが無効である。
ERR 6 Invalid input hole	指定された辺または穴の番号が無効である。
ERR 7 ALARM	技術スタッフ (technical staff) を呼んでください。

## 採点について

各箱に対し、箱の内容を可能な限り記述したテキストファイルを提出せよ。各箱に対し、

- あなたの解答で「.」、"/」、"\」のいずれかが間違った場所に書かれていたら、その箱に対するあなたの点数は 0 点となる。
- $B_m$  を提出されたすべての正しい解答の中で内容が決定されていたマス個数の最大値とし、 $B_y$  をあなたの解答の中で内容が決定されていたマス個数とすると、その箱に対するあなたの点数の割合は  $100B_y/B_m$  パーセントとなる。

注意 この課題に対する公式な解答では、これらの各箱の内容の 100% を 8 分未満でプログラムの的に決定することができる。