

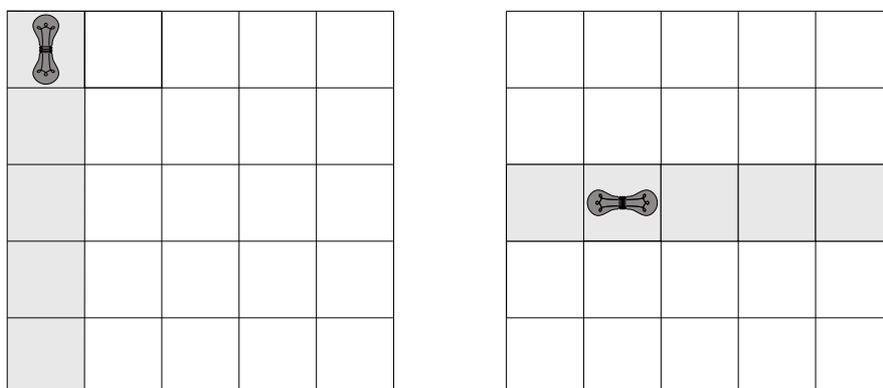
C. 電球 (Light Bulbs)

問題名	電球 (lightbulbs)
実行時間制限	4 秒
メモリ制限	1 GB

Frederik Philips 氏は、1891 年に Eindhoven 市の電球メーカーを設立した後、すごい発明をした。これは、縦方向または横方向に無限に光線が延びる電球である。彼はこの発明を使って、現代的な家の内装のデザインに革命を起こしたい。

いま、彼は息子の Gerard 氏とともに、電球の設置を入念に計画している。まず、部屋は $N \times N$ のマス目状の区画に区切られているが、各区画にひとつずつ、合計 N^2 個の電球を設置した。彼らは電力使用量を削減するため、最小の数の電球を ON にすることで部屋全体を明るくしたい。ここで、各電球の向きは縦方向 (同じ列のすべての区画を照らす) または横方向 (同じ行のすべての区画を照らす) のいずれかである。

縦方向の電球 (左) と横方向の電球 (右) の例を以下に示す。



しかし、残念ながら、彼らは電球の設置をする際に不注意であったため、各電球が縦方向であったか横方向であったかを覚えていない。このままでは、どの電球を ON にすると部屋全体が明るくなるのかわからない。そこで彼らは、何回かの実験を行うことにした。Gerard 氏は電球のある部屋にとどまり、Frederik 氏は別の部屋から電球のスイッチを操作する。

各実験では、Frederik 氏は各電球を ON にするか OFF にするかを決めて操作し、Gerard 氏は合計何個の区画が明るく照らされているかを報告する。ここで、2 つ以上の電球に照らされている区画についても「1 つ」としてだけカウントする。ここで、各実験では何個の電球を ON にしても問題ないが、彼らは急いでいるため、実験回数をできるだけ少なく抑えたい。

最小の数の電球を ON にすることで部屋全体を明るくする方法を見つけることで、彼らを手伝え。彼らは最大 2000 回の実験を行えるが、高得点を得るには、実験回数をより少なくしなければならない。

インタラクション

この問題はインタラクティブ問題である。

- あなたのプログラムは最初に整数 N を読み込む。 N は区画の縦の個数であり、区画の横の個数でもある。
- その後、あなたのプログラムは採点プログラムとやり取りを行う。 実験を行うには、最初に “?” からなる行を出力しなければならない。 そして次の N 行では、電球の ON/OFF に関する $N \times N$ の情報を出力しなければならない。 ここで出力する情報は、0 (OFF) と 1 (ON) からなる。 その後、照らされている区画の数を表す 1 つの整数 l ($0 \leq l \leq N^2$) を読み込まなければならない。
- 最後に答えを出力したいときは、最初に “!” からなる行を出力し、その後 N 行にわたって上と同じ形式で区画の電球の情報を出力しなければならない。 なお、正解 (Accepted) を出すためには、**電球はすべての区画を明るく照らす必要があります、なおかつ ON にする電球の数が最小でなければならない。**

その後、あなたのプログラムは終了しなければならない。

ここで、採点プログラムはアダプティブではない。 すなわち、各電球の向きはやり取りが始まる前に決まっている。

各出力の最後には、必ず標準出力を flush せよ。 さもなくば、Time Limit Exceeded と判定される可能性がある。 Python の場合、`input()` により自動的に flush される。 C++ の場合、`cout << endl;` によって、改行されるとともに自動的に flush される。 もし `printf` を使用する場合、`fflush(stdout)` を用いよ。

制約・採点形式

- $3 \leq N \leq 100$.
- あなたのプログラムは最大 2000 回の実験を行うことができる (最後の答えの出力は実験回数に含まれない)。 もし超過した場合、Wrong Answer と判定される。

あなたの解答は各小課題ごとに評価され、小課題にはそれぞれ配点が割り当てられている。 各小課題は複数のテストケースからなる。 各小課題について得点を得るためには、その小課題に含まれるすべてのテストケースに正解する必要がある。

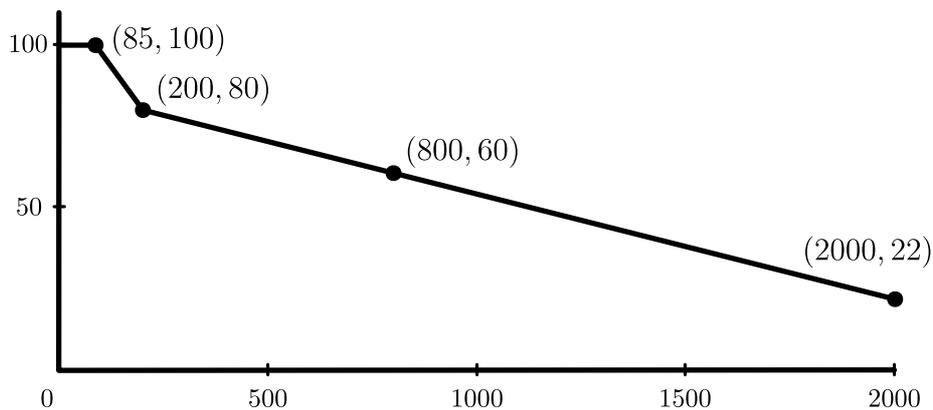
小課題	配点	制約
1	11	$N = 3$
2	11	$N \leq 10$
3	最大 78	追加の制約はない

最後の小課題では、あなたの得点は実験回数によって決まる。具体的には、以下の式によって決められる。

$$\text{score} = \begin{cases} (2000 - Q) \cdot 29/900 & \text{if } 200 \leq Q \leq 2000, \\ 58 + (200 - Q) \cdot 4/23 & \text{if } 85 \leq Q \leq 200, \\ 78 & \text{if } Q \leq 85. \end{cases}$$

ここで、 Q は実験回数が最も多かったテストケースにおける実験回数である。なお、得点は小数点以下切り捨てとなる。

以下のグラフは、すべてのテストケースで正解した場合における、 Q の値と得点の関係を示している。特に、この問題で 100 点を得るには、すべてのテストケースを 85 回以下の実験で解く必要がある。



テストのためのツール

あなたが解法をテストすることを容易にするため、簡単なツールがダウンロードできるようになっている。Kattis の問題ページ下部の "attachments" の項を見よ。ここで、本ツールを必ずしも使う必要はない。なお、Kattis で使用される実際の採点プログラムは、本ツールとは異なる。

本ツールを使うには、まず "sample1.in" のような入力ファイルを生成しなければならない。この入力ファイルでは、最初の行に整数 N 、次の N 行に各区画の情報を指定しなければならない。V は電球が縦方向であること、H は電球が横方向であることを意味する。具体例を以下に示す。

```
5
VVHVH
HVHHV
VHHVV
HHHVH
HHVVV
```

Python の場合、たとえば `solution.py` の場合、(通常は `pypy3 solution.py` とすれば実行できるが) 以下を実行しなければならない。

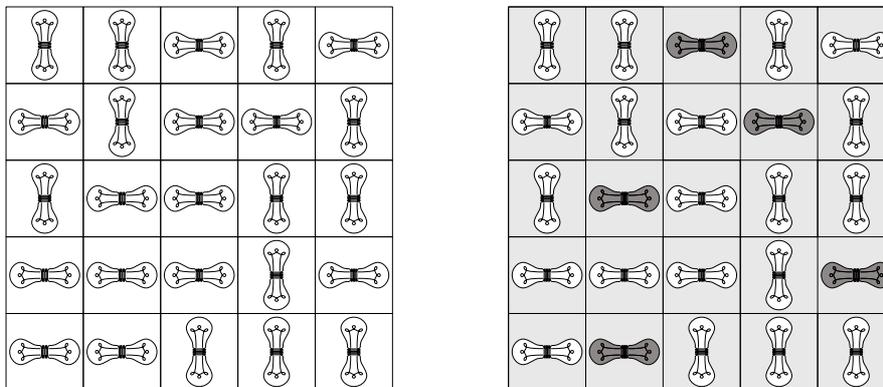
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

C++ の場合、まずコンパイルを行い(例: `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`), その後以下を実行しなければならない。

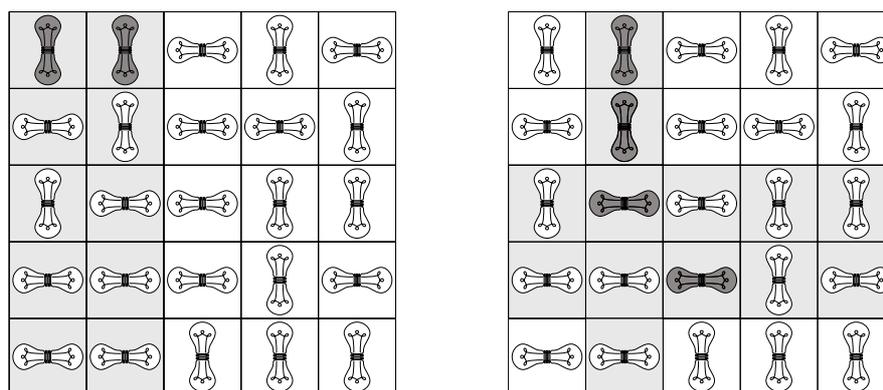
```
python3 testing_tool.py ./solution.out < sample1.in
```

例

サンプルテストケースでは、あなたのプログラムは最初に区画の大きさ $N = 5$ を読み込む。以下の図は、隠された各区画の情報(プログラムはまだ知らない)、および正解例のうち1つを示している。正解例では、5個の電球をONにすることで部屋全体を明るくしている。下図では、ONの電球は塗りつぶされており、照らされている区画は濃い色で表示されている。



このプログラムは下図に示す2回の実験を行う。1回目の実験では、左上の2個の縦方向の電球をONにすることで、10個の区画が照らされる。2回目の実験では13個の区画が照らされる。そして最後に、このプログラムは上図に示す答えを出力し、終了する。



採点プログラムの出力	あなたのプログラムの出力
5	
	? 11000 00000 00000 00000 00000
10	
	? 01000 01000 01000 00100 00000
13	
	! 00100 00010 01000 00001 01000