

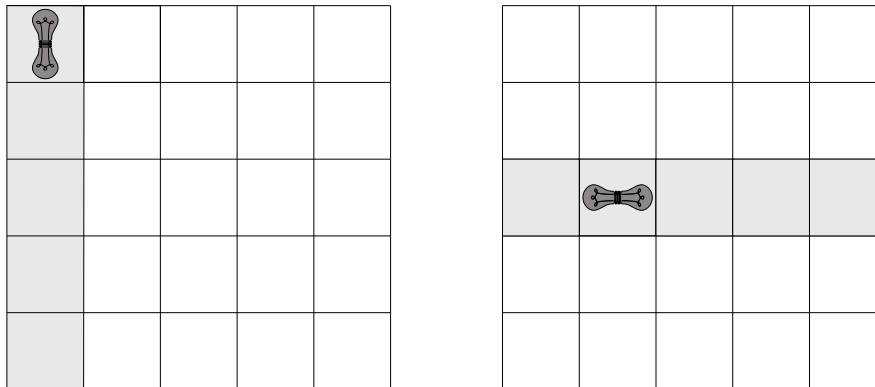
C. Glühbirnen

| Aufgabenname | lightbulbs |
|--------------|------------|
| Time Limit | 4 Sekunden |
| Memory Limit | 1 Gigabyte |

Kurz nach der Gründung seiner Glühbirnenfirma in Eindhoven im Jahr 1891 machte Frederik Philips eine grosse Entdeckung: Glühbirnen die einen unendlichen Strahl in horizontaler oder vertikaler Richtung ausleuchten. Mit dieser neuen Entdeckung will er die Inneneinrichtung moderner Häuser revolutionieren.

Zusammen mit seinem Sohn Gerard plant er eine aufwendige Installation. Sie installieren N^2 Lampen in einem $N \times N$ Gitter in einem Raum. Sie wollen den ganzen Raum mit so wenigen Lampen wie möglich beleuchten, um Strom zu sparen. Jede Lampe ist entweder vertikal, d.h. sie beleuchtet alle Felder in ihrer Spalte, oder horizontal, d.h. sie beleuchtet alle Felder in ihrer Reihe.

Die folgende Abbildung zeigt ein Beispiel für eine vertikale (links) und eine horizontale (rechts) Lampe.



Leider haben sie bei der Installation der Lampen nicht darauf geachtet und wissen nicht mehr, welche Lampen waagrecht oder senkrecht leuchten. Stattdessen führen sie einige Experimente durch, um herauszufinden, mit welchen Lampen sie den ganzen Raum beleuchten können. Gerard bleibt in dem Raum mit den Lampen, während Frederik von einem anderen Raum aus die Schalter bedient.

Bei jedem Experiment schaltet Frederik jede Lampen ein oder aus, und Gerard berichtet, wie viele Quadrate insgesamt beleuchtet sind; ein Quadrat, das von zwei oder mehr verschiedenen Lampen

beleuchtet wird, wird nur einmal gezählt. Es spielt keine Rolle, wie viele Lampen bei den Experimenten eingeschaltet werden, aber sie haben es eilig und wollen im Idealfall so wenige Experimente wie möglich durchführen.

Hilf ihnen, eine Anordnung von Lampen zu finden, die den ganzen Raum ausleuchtet und die wenigsten Lampen benötigt. Sie können höchstens 2000 Experimente durchführen. Du erhältst jedoch eine höhere Punktzahl, wenn sie weniger Experimente durchführen.

Interaktion

Dies ist ein interaktives Problem.

- Dein Programm sollte damit beginnen, eine Zeile mit einer ganzen Zahl N zu lesen, die die Höhe und Breite des Rasters angibt.
- Dann sollte dein Programm mit dem Grader interagieren. Um ein Experiment durchzuführen, solltest du zuerst eine Zeile mit einem Fragezeichen „?“ ausgeben. Schreibe in die folgenden N Zeilen ein $N \times N$ Gitter aus 0en und 1en, die angeben, welche Lampen aus (0) oder an (1) sein sollen. Dann sollte dein Programm eine einzelne ganze Zahl ℓ ($0 \leq \ell \leq N^2$) lesen, die Anzahl der Gitterquadrate, die durch das Einschalten der angegebenen Lampen erleuchtet werden.
- Wenn du antworten willst, gib eine Zeile mit einem Ausrufezeichen „!“ aus, gefolgt von N Zeilen mit dem Gitter in demselben Format wie oben. Damit deine Antwort akzeptiert wird, müssen die **Lampen das gesamte Gitter beleuchten und die Anzahl der eingeschalteten Lampen muss so gering wie möglich sein.**

Danach sollte sich dein Programm beenden.

Der Grader ist nicht adaptiv, d.h. das Raster der Lampen wird vor Beginn der Interaktion festgelegt.

Stelle sicher, dass die Standardausgabe nach jedem Experiment geflushet wird; andernfalls könnte dein Programm als „Time Limit Exceeded“ bewertet werden. In Python geschieht dies automatisch, solange du `input()` verwendest, um Zeilen zu lesen. In C++ wird mit `cout << endl;` zusätzlich zur Ausgabe eines Zeilenumbruchs ein Flush durchgeführt; wenn du `printf` verwendest, verwende `fflush(stdout)`.

Einschränkungen und Bewertung

- $3 \leq N \leq 100$.
- Du kannst höchstens 2000 Experimente durchführen (das Ausgeben der endgültigen Antwort zählt nicht als ein Experiment). Wenn du diese Zahl überschreitest, erhältst du die Rückmeldung „Wrong Answer“.

Deine Lösung wird auf einer Reihe von Testgruppen getestet, welche eine gewisse Anzahl an Punkten wert sind. Jede Testgruppe enthält eine Reihe von Testfällen. Um die Punkte für eine Testgruppe zu erhalten, musst du alle Testfälle der Testgruppe lösen.

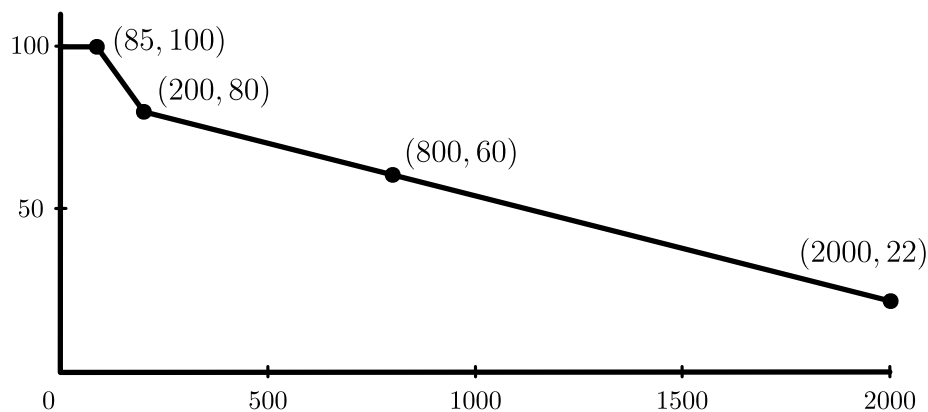
| Gruppe | Punkte | Limits |
|--------|-----------|--------------------------------|
| 1 | 11 | $N = 3$ |
| 2 | 11 | $N \leq 10$ |
| 3 | bis zu 78 | Keine weiteren Einschränkungen |

In der letzten Testgruppe hängt deine **Punktzahl von der Anzahl der von dir durchgeführten Experimente** ab, die nach der folgenden Formel berechnet wird:

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

wobei Q die maximale Anzahl von Experimenten ist, die für einen Testfall verwendet werden. Die Punktzahl wird auf die nächste ganze Zahl abgerundet.

Das folgende Diagramm zeigt die Anzahl der Punkte als Funktion von Q , die dein Programm erhält, wenn es alle Testgruppen löst. Um die volle Punktzahl von 100 Punkten zu erreichen, musst du jeden Testfall mit höchstens 85 Experimenten lösen.



Testtool

Um das Testen deiner Lösung zu erleichtern, stellen wir dir ein einfaches Tool zur Verfügung, das du herunterladen kannst. Siehe „attachments“ am Ende der Kattis-Problempage. Die Verwendung des Tools ist optional. Beachte, dass das offizielle Grader auf Kattis sich von dem Testtool unterscheidet.

Um das Tool zu benutzen, erstelle eine Eingabedatei, wie z.B. „sample1.in“, die mit einer Zahl N beginnen sollte, gefolgt von N Zeilen, die das Gitter spezifizieren, wobei v bedeutet, dass die

Lampe ihre Spalte beleuchtet und H bedeutet, dass sie ihre Zeile beleuchtet. Zum Beispiel:

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

Bei Python-Programmen wird `solution.py` genutzt (normalerweise als `pypy3 solution.py` ausgeführt):

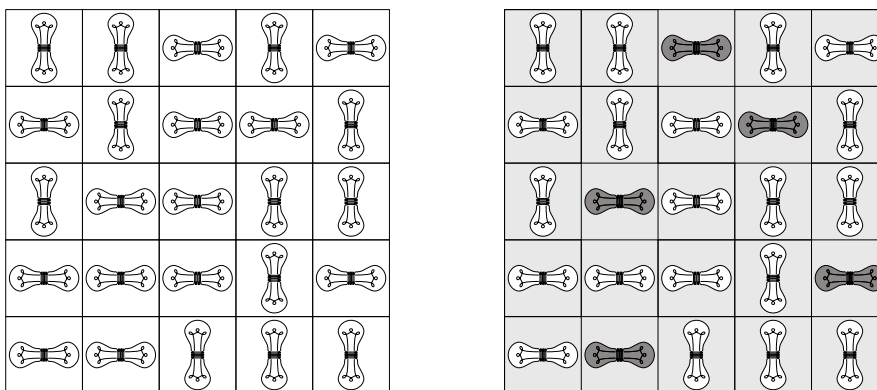
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Bei C++-Programmen kompiliere es zuerst (z.B. mit `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) und führe es dann aus:

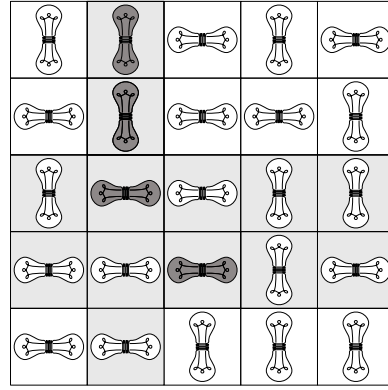
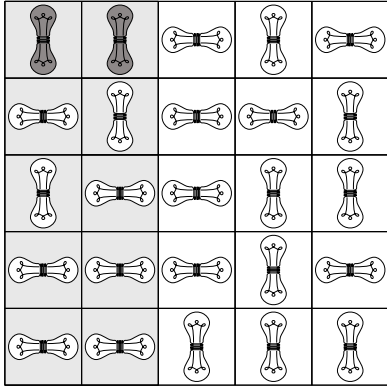
```
python3 testing_tool.py ./solution.out < sample1.in
```

Beispiel

In der Beispielinteraktion beginnt das Programm mit dem Lesen der Gittergröße $N = 5$. Die folgende Abbildung zeigt das verborgene Gitter (welches das Programm nicht kennt) und eine von vielen möglichen Antworten, wobei fünf Lampen das gesamte Gitter beleuchten. Die markierten Lampen sind eingeschaltet und die dunkleren Quadrate sind beleuchtet.



Das Programm führt zwei Experimente durch, wie unten dargestellt. Im ersten Experiment werden insgesamt 10 Quadrate mit Hilfe der beiden vertikalen Lampen in der oberen linken Ecke beleuchtet. Beim zweiten Experiment leuchten die Lampen insgesamt 13 Quadrate aus. Schliesslich schreibt das Programm seine Antwort (siehe oben) und beendet sich.



| Graderausgabe | Deine Ausgabe |
|---------------|--|
| 5 | |
| | ? 11000 00000 00000 00000 00000 |
| 10 | |
| | ? 01000 01000 01000 00100 00000 |
| 13 | |
| | ! 00100 00010 01000 00001 01000 |