

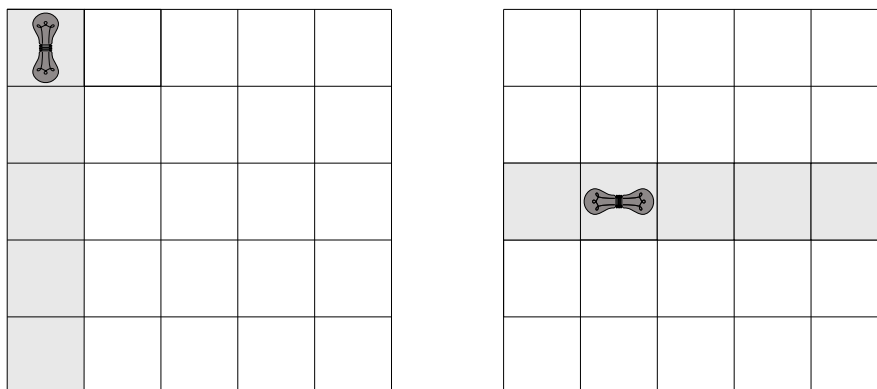
C. Żarówki

Nazwa zadania	lightbulbs
Limit czasu	4 sekundy
Limit pamięci	1 GB

Jak wiadomo, Frederik Philips założył firmę żarówkową w Eindhoven w 1891 roku. Teraz jego śladami podążają Krzysztof i Monika, którzy także założyli firmę żarówkową. Niedawno dokonali wielkiego odkrycia: żarówki, które rozświetlają nieskończony promień w wybranym kierunku - pionowym lub poziomym. Dzięki temu wynalazkowi chcą zrewolucjonizować wystrój współczesnych domów.

Chcą zainstalować N^2 lamp na siatce o wymiarach $N \times N$ reprezentującej pokój. Chcą rozświetlić cały pokój, używając jak najmniej włączonych lamp, aby oszczędzać prąd. Lampa może być pozioma, wówczas oświetla wszystkie pola w swoim rzędzie, lub pionowa, wówczas oświetla wszystkie pola w swojej kolumnie.

Ilustracja poniżej pokazuje przykład lampy pionowej (z lewej strony) i poziomej (z prawej strony).



Niestety nie uważali wystarczająco, kiedy zakładali lampy i nie pamiętają, które lampy świecą się poziomo, a które pionowo. Zamiast tego przeprowadzili eksperymenty, aby dowiedzieć się, których lamp użyć, by oświetlić cały pokój. Monika stoi w pokoju z lampami, kiedy Krzysztof operuje przełącznikami z innego pomieszczenia.

W każdym eksperymencie Krzysztof włącza pewne lampy i Monika zgłasza, ile pól jest oświetlonych. Pole oświetlone przez dwie lub więcej lamp jest liczone jednokrotnie. Nie ma

znaczenia, ile lamp jest włączonych w czasie eksperymentu, ale Monika i Krzysztof bardzo się śpieszą, więc chcą przeprowadzić mało eksperymentów.

Pomóż im znaleźć ustawienie lamp, które rozświetla cały pokój i używa minimalnej liczby lamp. Mogą przeprowadzić co najwyżej 2 000 eksperymentów. Jednak dostaniesz więcej punktów, jeśli przeprowadzą mniej eksperymentów.

Interakcja

To jest zadanie interaktywne.

- Twój program powinien rozpocząć od przeczytania linii zawierającej liczbę całkowitą N oznaczającą wysokość i szerokość siatki.
- Potem Twój program powinien się komunikować ze sprawdzaczką. Żeby przeprowadzić eksperyment, Twój program powinien najpierw wypisać znak zapytania "?". W kolejnych N liniach powinnaś wypisać siatkę $N \times N$ zer i jedynek oznaczających, która lampa powinna być wyłączona (0) lub włączona (1). Potem Twój program powinien wczytać jedną liczbę całkowitą ℓ ($0 \leq \ell \leq N^2$) - liczbę pól siatki rozświetlonych przez te lampy.
- Kiedy chcesz zwrócić odpowiedź, wypisz linię z wykrzyknikiem "!", a po niej N linii reprezentujących siatkę w takim samym formacie jak wyżej. Aby Twoja odpowiedź została zaakceptowana, **lampy muszą rozświetlać całą siatkę i liczba włączonych lamp musi być najmniejsza możliwa.**

Potem Twój program powinien zakończyć swoje działanie.

Sprawdzaczka jest nieadaptacyjna, to znaczy siatka lamp jest wyznaczana, zanim zaczyna się interakcja.

Upewnij się, żeby wyczyścić bufor standardowego wyjścia po wykonaniu każdego eksperymentu. W przeciwnym przypadku Twój program może dostać werdykt "Time Limit Exceeded". W Pythonie dzieje się to automatycznie, jeśli używasz `input()` do wczytywania linii. W C++ `cout << endl;` czyści bufor i wypisuje znak nowej linii; jeśli korzystasz z `printf`, użyj `fflush(stdout)`.

Podzadania i ocenianie

- $3 \leq N \leq 100$.
- Możesz wykonać co najwyżej 2 000 eksperymentów (wypisanie ostatecznej odpowiedzi nie liczy się jako eksperyment). Jeśli przekroczysz tę liczbę, dostaniesz werdykt "Wrong Answer".

Twoje rozwiązanie będzie testowane na zbiorze grup testów, każda warta pewną liczbę punktów. Każda grupa testów zawiera zbiór przypadków testowych. Żeby dostać punkty za grupę testów, musisz rozwiązać wszystkie przypadki testowe w danej grupie testów.

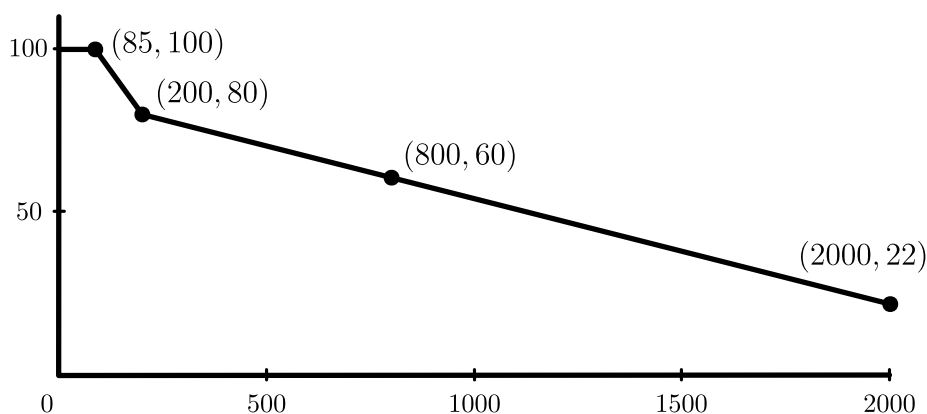
Grupa	Punktacja	Ograniczenia
1	11	$N = 3$
2	11	$N \leq 10$
3	do 78	Brak dodatkowych ograniczeń

W ostatniej grupie testów **Twój wynik zależy od liczby eksperymentów, które przeprowadzisz** i jest obliczany według następującego wzoru:

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

gdzie Q jest maksymalną liczbą eksperymentów przeprowadzonych we wszystkich przypadkach testowych. Wynik będzie zaokrąglony w dół do najbliższej liczby całkowitej.

Wykres poniżej ukazuje liczbę punktów, które Twój program otrzyma, gdy rozwiąże wszystkie grupy testów, jako funkcję Q . Aby otrzymać pełny wyniki 100 punktów za to zadanie, musisz rozwiązać każdy przypadek testowy, używając maksymalnie 85 eksperymentów.



Narzędzie do testowania

Aby uprościć testowanie Twojego rozwiązania, zapewniamy proste narzędzie, które możesz pobrać. Zobacz sekcję "attachments" na dole strony Kattis. Używanie narzędzia jest opcjonalne. Zauważ, że oficjalne narzędzie do oceniania na Kattis jest inne od narzędzia do testowania.

Żeby użyć narzędzia, stwórz plik wejściowy, na przykład "sample1.in", który powinien zaczynać się liczbą N , po której następuje N linii specyfikujących siatkę, gdzie v oznacza, że lampa rozświetla swoją kolumnę i H oznacza, że rozświetla swój rząd. Przykładowo:

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

Dla programów w Pythonie, np. `solution.py` (normalnie uruchamiany jako `pypy3 solution.py`):

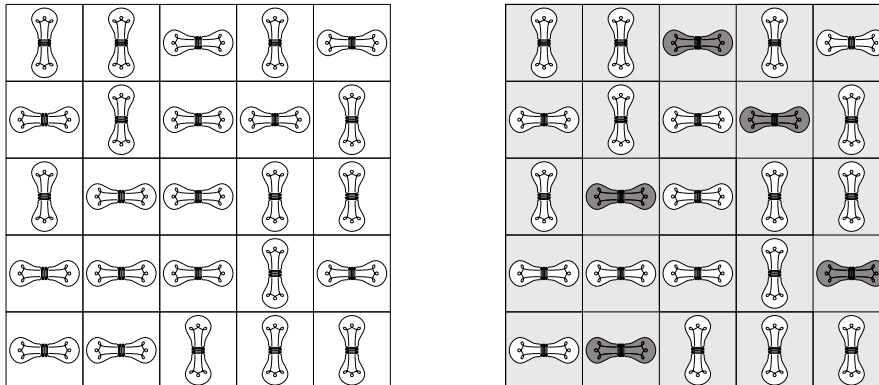
```
python3 testing_tool.py pypy3 solution.py < sample1.in
```

Programy w C++ należy najpierw skompilować (np. poleceniem `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`), a potem uruchomić:

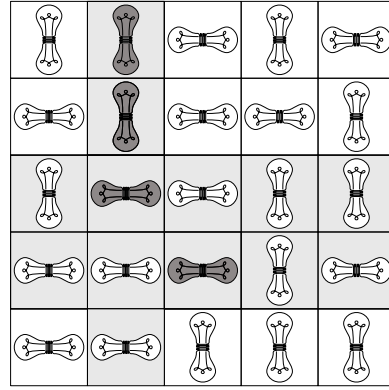
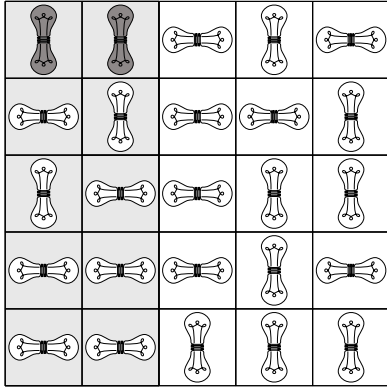
```
python3 testing_tool.py ./solution.out < sample1.in
```

Przykład

W przykładowej interakcji program zaczyna od czytania rozmiaru siatki $N = 5$. Następujący obrazek pokazuje ukrytą siatkę (której program nie zna) i jedną z wielu możliwych odpowiedzi używającą pięciu lamp do rozświetlenia całej siatki. Zaznaczone lampy są włączone i ciemniejsze pola są rozświetlone.



Program wykonuje dwa eksperymenty jak przedstawiono poniżej. W pierwszym eksperymencie łącznie 10 pól jest rozświetlonych używając dwóch lamp pionowych w lewym górnym rogu. Drugi eksperyment rozświetla łącznie 13 pól. Ostatecznie program wypisuje swoją odpowiedź (zilustrowaną powyżej) i kończy swoje działanie.



wyjście sprawdzaczki	Twoje wyjście
5	
	? 11000 00000 00000 00000 00000
10	
	? 01000 01000 01000 00100 00000
13	
	! 00100 00010 01000 00001 01000