

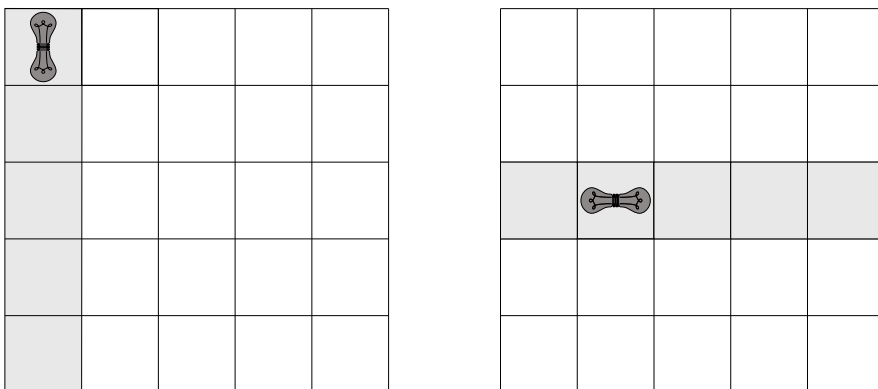
## C. Γλόμπιοι (lightbulbs)

Όνομα προβλήματος	Γλόμπιοι
Χρονικό Όριο	4 δευτερόλεπτα
Όριο Μνήμης	1 gigabyte

Λίγο μετά την ίδρυση της εταιρίας γλόμπων του στο Eindhoven το 1891, ο Frederik Philips έκανε μια σπουδαία ανακάλυψη: γλόμπους που φωτίζουν μια άπειρη ακτίνα σε οριζόντια ή κάθετη κατεύθυνση. Με αυτή τη νέα ανακάλυψη, θέλει να φέρει επανάσταση στον εσωτερικό σχεδιασμό των σύγχρονων σπιτιών.

Σχεδιάζει μια περίτεχνη εγκατάσταση με τον γιο του, τον Gerald. Εγκαθιστούν  $N^2$  γλόμπους στο  $N \times N$  πλέγμα ενός δωματίου. Θέλουν να φωτίσουν όλο το δωμάτιο με όσο το δυνατόν λιγότερους αναμμένους γλόμπους για να εξοικονομήσουν ηλεκτρική ενέργεια. Κάθε γλόμπος είναι είτε κάθετος, δηλαδή φωτίζει όλα τα τετράγωνα στη στήλη του, είτε οριζόντιος, δηλαδή φωτίζει όλα τα τετράγωνα στη σειρά του.

Η παρακάτω εικόνα δείχνει ένα παράδειγμα κατακόρυφου (αριστερά) και οριζόντιου (δεξιά) γλόμπου.



Δυστυχώς, δεν έδωσαν προσοχή καθώς εγκαθιστούσαν τους γλόμπους και δεν θυμούνται ποιοι γλόμπιοι ανάβουν οριζόντια ή κάθετα. Αντ' αυτού, διεξάγουν κάποια πειράματα για να καταλάβουν ποιους γλόμπους να χρησιμοποιήσουν για να φωτίσουν ολόκληρο το δωμάτιο. Ο Gerald παραμένει στο δωμάτιο με τους γλόμπους, ενώ ο Frederik χειρίζεται τους διακόπτες από ένα άλλο δωμάτιο.

Σε κάθε πείραμα, ο Frederik ανάβει ή σβήνει κάθε γλόμπο και ο Gerald αναφέρει πόσα τετράγωνα φωτίζονται συνολικά: ένα τετράγωνο που φωτίζεται από δύο ή περισσότερους ξεχωριστούς γλόμπους μετράται μόνο μία φορά. Δεν έχει σημασία πόσοι γλόμποι ανάβουν κατά τη διάρκεια των πειραμάτων, αλλά βιάζονται και ιδανικά θέλουν να διεξάγουν όσο το δυνατόν λιγότερα πειράματα.

Βοηθήστε τους να βρουν μια διάταξη γλόμπων που να φωτίζει όλο το δωμάτιο και να χρησιμοποιεί τους λιγότερους αναμμένους γλόμπους. Μπορούν να πραγματοποιήσουν το πολύ 2000 πειράματα. Ωστόσο, θα λάβουν υψηλότερη βαθμολογία αν χρησιμοποιήσουν λιγότερα πειράματα.

## Αλληλεπίδραση (Interaction)

Αυτό είναι ένα διαδραστικό πρόβλημα.

- Το πρόγραμμά σας θα πρέπει να ξεκινήσει διαβάζοντας μια γραμμή με έναν ακέραιο αριθμό  $N$ , το ύψος και το πλάτος του πλέγματος.
- Στη συνέχεια, το πρόγραμμά σας θα πρέπει να αλληλεπιδράσει με τον βαθμολογητή. Για να πραγματοποιήσετε ένα πείραμα, θα πρέπει πρώτα να εκτυπώσετε μια γραμμή με ένα ερωτηματικό "?". Στις επόμενες  $N$  γραμμές, τυπώστε ένα πλέγμα  $N \times N$  που καθορίζει ποιοι γλόμποι είναι αναμμένοι. Συγκεκριμένα, σε καθεμία από αυτές τις γραμμές, τυπώστε μια συμβολοσειρά μήκους  $N$ , αποτελούμενη από 0 (για τους σβηστούς) και 1 (για τους αναμμένους) γλόμπους. Στη συνέχεια, το πρόγραμμά σας θα πρέπει να διαβάζει έναν μόνο ακέραιο  $\ell$  ( $0 \leq \ell \leq N^2$ ), το πλήθος των τετραγώνων του πλέγματος που φωτίζονται με το άναμμα των γλόμπων που έχουν καθοριστεί.
- Όταν θέλετε να απαντήσετε, εκτυπώστε μια γραμμή με θαυμαστικό "!", ακολουθούμενη από  $N$  γραμμές με το πλέγμα στην ίδια μορφή όπως παραπάνω. Για να γίνει δεκτή η απάντησή σας, **οι γλόμποι πρέπει να φωτίζουν όλο το πλέγμα και ο αριθμός των ενεργοποιημένων γλόμπων πρέπει να είναι ο ελάχιστος δυνατός.**

Έπειτα το πρόγραμμά σας θα πρέπει να τερματίσει.

Ο βαθμολογητής είναι μη προσαρμοστικός (not adaptive), που σημαίνει ότι το πλέγμα με τους γλόμπους καθορίζεται πριν αρχίσει η αλληλεπίδραση.

Βεβαιωθείτε να κάνετε flush την τυπική έξοδο (standard output) μετά την περιγραφή κάθε πειράματος: διαφορετικά, το πρόγραμμά σας μπορεί να τερματιστεί με ετυμηγορία "Time Limit Exceeded". Στην Python, αυτό συμβαίνει αυτόματα εφόσον χρησιμοποιείτε την εντολή `input()` για να διαβάσετε γραμμές. Στη C++, η εντολή `cout << endl;` κάνει flush ενώ επιπλέον τυπώνει και έναν χαρακτήρα νέας γραμμής; αν χρησιμοποιείτε `printf`, χρησιμοποιήστε το `fflush(stdout)`.

## Περιορισμοί και Βαθμολόγηση

- $3 \leq N \leq 100$ .
- Μπορείτε να διεξάγετε το πολύ 2 000 πειράματα (η εκτύπωση της τελικής απάντησης δεν μετράει ως πείραμα). Αν το υπερβείτε, θα λάβετε την ετυμηγορία "Wrong Answer".

Η λύση σας θα δοκιμαστεί σε ένα σύνολο ομάδων δοκιμών (test groups), καθεμία από τις οποίες παίρνει κάποιους βαθμούς. Κάθε ομάδα δοκιμών περιέχει ένα σύνολο δοκιμαστικών περιπτώσεων (test cases). Για να λάβετε τους βαθμούς για μια ομάδα δοκιμών, πρέπει να επιλύσετε όλες τις δοκιμαστικές περιπτώσεις της ομάδας δοκιμών.

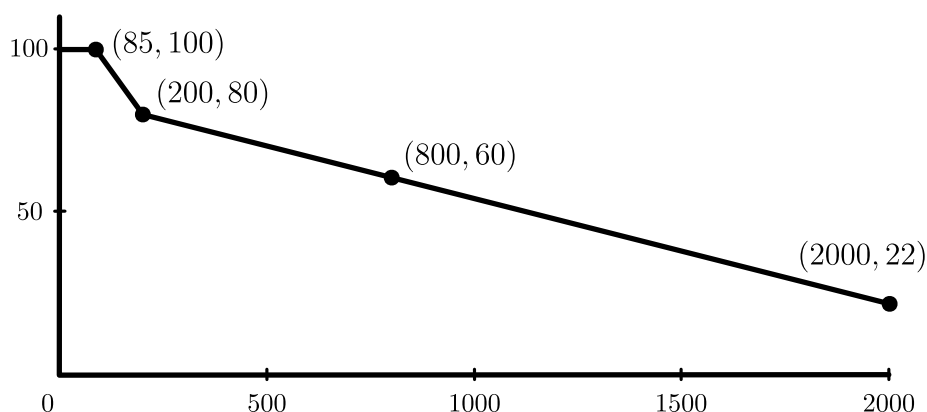
Ομάδα	Βαθμολογία	Περιορισμοί
1	11	$N = 3$
2	11	$N \leq 10$
3	έως 78	Χωρίς επιπλέον περιορισμούς

Στην τελική ομάδα δοκιμών, η **βαθμολογία σας εξαρτάται από τον αριθμό των πειραμάτων που διεξάγετε** και υπολογίζεται από τον ακόλουθο τύπο:

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

όπου  $Q$  είναι ο μέγιστος αριθμός πειραμάτων που χρησιμοποιούνται σε κάθε testcase. Η βαθμολογία στρογγυλοποιείται στον πλησιέστερο ακέραιο αριθμό.

Το παρακάτω διάγραμμα δείχνει τον αριθμό των βαθμών, ως συνάρτηση του  $Q$ , που θα πάρει το πρόγραμμά σας αν λύσει όλες τις ομάδες δοκιμών. Για να λάβετε την πλήρη βαθμολογία σε αυτό το πρόβλημα, πρέπει να λύσετε κάθε περίπτωση δοκιμής χρησιμοποιώντας το πολύ 85 πειράματα.



## Εργαλείο Δοκιμής

Για να διευκολυνθεί η δοκιμή της λύσης σας, παρέχουμε ένα απλό εργαλείο που μπορείτε να κατεβάσετε. Δείτε τα «συνημμένα» στο κάτω μέρος της σελίδας του προβλήματος στο Kattis. Η χρήση του εργαλείου είναι προαιρετική. Σημειώστε ότι το επίσημο πρόγραμμα βαθμολόγησης στο Kattis είναι διαφορετικό από το εργαλείο δοκιμής.

Για να χρησιμοποιήσετε το εργαλείο, δημιουργήστε ένα αρχείο εισόδου, όπως το "sample1.in", το οποίο θα πρέπει να ξεκινά με έναν αριθμό  $N$  ακολουθούμενο από  $N$  γραμμές που καθορίζουν το πλέγμα, όπου  $v$  σημαίνει ότι ο γλόμπος φωτίζει τη στήλη του και  $h$  σημαίνει ότι φωτίζει τη γραμμή του. Για παράδειγμα:

```
5
VHVHV
HVHHV
VHHVV
HHHVV
HHVVV
```

Για προγράμματα Python, όπως `solution.py` (που συνήθως τρέχουν με την εντολή `python3 solution.py`):

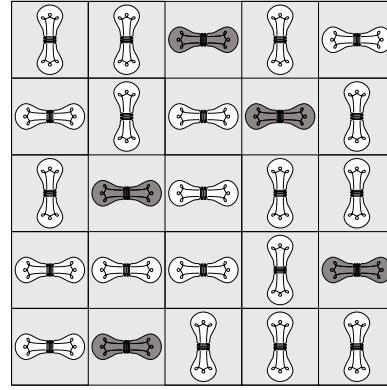
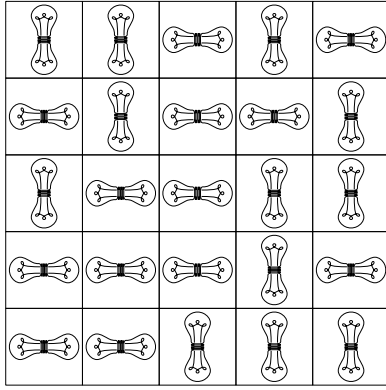
```
python3 testing_tool.py python3 solution.py < sample1.in
```

Για προγράμματα C++, πρώτα να κάνετε μεταγλώττιση (compile) (π.χ. τρέχοντας `g++ -g -O2 -std=gnu++20 -static solution.cpp -o solution.out`) και μετά τρέξτε:

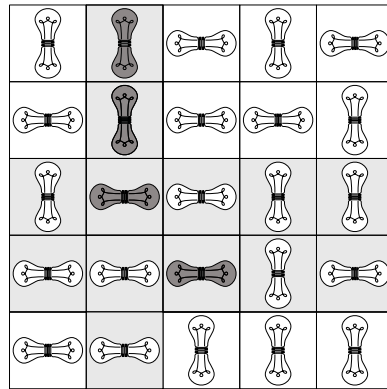
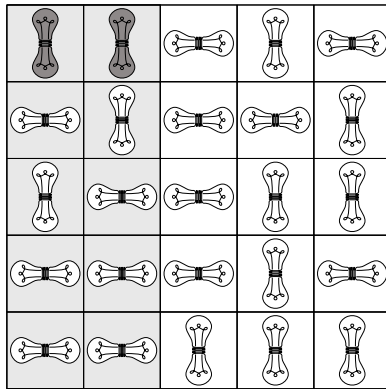
```
python3 testing_tool.py ./solution.out < sample1.in
```

## Παράδειγμα

Στο αλληλεπιδραστικό παράδειγμα, το πρόγραμμα ξεκινά διαβάζοντας το μέγεθος πλέγματος  $N = 5$ . Το παρακάτω σχήμα δείχνει το κρυφό πλέγμα (το οποίο το πρόγραμμα δεν γνωρίζει) και μία από τις πολλές πιθανές απαντήσεις, χρησιμοποιώντας πέντε γλόμπους για να φωτίσει ολόκληρο το πλέγμα. Οι μαρκαρισμένοι γλόμπους είναι αναμμένοι και τα πιο σκούρα τετράγωνα φωτίζονται.



Το πρόγραμμα εκτελεί δύο πειράματα όπως φαίνεται παρακάτω. Στο πρώτο πείραμα, φωτίζονται συνολικά 10 τετράγωνα χρησιμοποιώντας τους δύο κάθετους γλόμπους στην επάνω δεξιά γωνία. Στο δεύτερο πείραμα φωτίζονται συνολικά 13 τετράγωνα. Τέλος, το πρόγραμμα γράφει την απάντησή του (που απεικονίζεται παραπάνω) και τερματίζει.



δεδομένα έξοδου βαθμολογική	δεδομένα εξόδου προγράμματος
5	
	? 11000 00000 00000 00000 00000
10	
	? 01000 01000 01000 00100 00000
13	
	! 00100 00010 01000 00001 01000