

Leuchte!

Wer etwas auf sich hält, kann gar nicht früh genug damit anfangen, sein Haus weihnachtlich herauszuputzen. Unglücklicherweise sind nur wenige Dinge

- (a) der geistigen Gesundheit und
- (b) jedweder Form zwischenmenschlicher Beziehung

so abträglich wie das Schmücken von Weihnachtsbäumen. Vor allem, wenn mal wieder irgendetwas nicht funktioniert.

Walter Hoppenstedt hat in weiser Voraussicht und aus Rücksichtnahme auf (b) bereits alle Familienmitglieder zum Weihnachtseinkauf geschickt und damit aus dem näheren Umkreis verbannt. Allerdings muss er sich dieser Aufgabe daher nun gänzlich alleine stellen—und (a) droht akute Gefahr.

In einer bemerkenswert akrobatischen Aktion hat Walter bereits die Lichterkette mit n ($1 \leq n \leq 1\,000$) Lampen in symmetrischer und ästhetisch ansprechender Weise an der Tanne im Garten befestigt und das Kabel elegant zur Steckdose im Keller verlegt. Doch nach Einstecken des Steckers muss er feststellen, dass einige der Lampen nicht brennen.

Dafür, dass eines der Lichter nicht brennt, kann es im Wesentlichen zwei Gründe geben: entweder die Birne ist kaputt oder die Fassung. Walter möchte nun die Lampen so auf die Fassungen verteilen, dass möglichst viele brennen. Da er allerdings gerade kein Messgerät zur Hand ist, besteht die einzige Möglichkeit, Informationen über den Zustand der Lichterkette zu gewinnen, darin, einige der Birnen in andere Fassungen zu schrauben.

Da die Lichterkette dazu natürlich erst ausgesteckt (niemand möchte einen Stromschlag bekommen) und dann danach wieder eingesteckt werden muss (was erfordert, zweimal den langen Weg in den Keller auf sich zu nehmen), können wir annehmen, dass die Zeit für einen solchen Versuch unabhängig von der Anzahl der umgeschraubten Lampen ist. Allerdings bleiben Walter nur noch 1 500 Versuche, bis der Rest der Familie wieder zuhause aufschlägt und er seine Schmach eingestehen müsste. Hilf ihm, indem du ein Programm schreibst, das für ihn ermittelt, welche Lampen er in welche Fassung schrauben soll, damit maximal viele Birnen leuchten!

Anmerkung

Die Lichterkette des Anstoßes ist glücklicherweise ein moderneres Modell—das heißt, nur weil eine Glühbirne oder Fassung defekt ist, bleiben nicht auch automatisch die restlichen Kerzen finster. Mit anderen Worten: eine Kerze brennt genau dann, wenn weder Fassung noch Birne kaputt sind.

Implementierung

Dies ist eine interaktive Aufgabe. Dein Programm bekommt zunächst n , die Anzahl der Lichter, in einer Zeile gegeben. Danach kannst du eine der folgenden Anweisungen geben, indem du eine entsprechende Zeile ausgibst. Anschließend kannst du (falls vorhanden) die Antwort auf die Anweisung einlesen. Die Anweisungen sind:

- **Q** $x_1 \dots x_n$, wobei die x_i paarweise verschiedene ganze Zahlen zwischen 1 und n sind, gibt die Anweisung, für $i = 1, \dots, n$ in Fassung i die x_i -te Glühbirne zu schrauben. Als Antwort erhältst du eine Zeile mit n ganzen Zahlen. Hierbei ist die i -te Zahl eine 1, wenn die Birne in der i -ten Fassung leuchtet, sonst eine 0.

- **A** $x_1 \dots x_n$ (mit derselben Beschränkung wie oben) darf nur ein einziges Mal ausgegeben werden—dein Programm wird danach automatisch beendet. Die Anweisung bedeutet, dass du der Ansicht bist, dass die angegebene Anordnung der Glühbirnen optimal ist.

Dein Programm darf höchstens 1 500 Anfragen vom ersten Typ stellen. Bei einer Ausgabe, die nicht den obigen Bedingungen gehorcht, wird deine Einsendung für den entsprechenden Testfall als falsch gewertet.

Wichtiger Hinweis

Um eine reibungslose Kommunikation zu gewährleisten, musst du nach jeder deiner Ausgaben die Standardausgabe »flushen«. Wenn du `cout` verwendest, genügt es, am Ende jeder Ausgabe `<< flush` anzufügen, also z.B.

`cout << "Q 1 2 3 4\n" << flush;`

oder generell `endl` für Zeilenumbrüche zu verwenden. Bei Verwendung von `printf` musst du nach jeder Ausgabe `fflush(stdout)` aufrufen.

Beschränkungen und Bewertung

Stets gilt $1 \leq n \leq 1\,000$.

Teilaufgabe 1 (20 Punkte). $n \leq 5$

Teilaufgabe 2 (20 Punkte). $n \leq 30$

Teilaufgabe 3 (40 Punkte). $n \leq 400$

Teilaufgabe 4 (20 Punkte). Keine weiteren Beschränkungen.

Beispielinteraktion

Im öffentlichen Testfall besteht die Lichterkette aus $n = 4$ Lichtern, wobei genau die ersten beiden Fassungen und die letzten beiden Glühbirnen kaputt sind.

Dein Programm	Grader	Erklärung
	4	Anzahl der Glühbirnen
Q 1 2 3 4	0 0 0 0	kein Licht brennt
Q 4 3 2 1	0 0 1 1	die letzten beiden Lichter brennen
A 4 3 2 1		dein Programm möchte die Lösung ausgeben die Lösung ist korrekt und wird akzeptiert

Limits

Zeit: 1.5 s

Speicher: 256 MiB

Das obige Zeitlimit enthält auch die Laufzeit des Graders; es ist allerdings garantiert, dass das Beantworten der 1 500 erlaubten Anfragen höchstens die Hälfte davon benötigt.

Feedback

Für diese Aufgabe ist *restricted feedback* verfügbar. Das bedeutet, die angezeigte Punktzahl entspricht der endgültigen Punktzahl deiner Einsendung. Allerdings wird dir für jede Testfallgruppe immer nur der erste Testfall mit minimaler Punktzahl innerhalb der entsprechenden Gruppe angezeigt. (Hierbei ist die Reihenfolge der Fälle innerhalb der jeweiligen Gruppen fest.)