

Hühnerhof

Nachdem Florian nun schon zu alt ist, um an der IOI teilnehmen zu können, möchte er sich jetzt auf Hühnerzucht spezialisieren.

Der Jungbauer hat bereits n Hühner auf seinem Hof, welche von 1 bis n nummeriert sind. Diese picken einander laufend, obwohl ihnen Florian eigentlich reichlich Futter gibt. Wahrscheinlich müssen sich die Hühner einfach noch an ihre neuen Mitbewohner gewöhnen.

Das Picken ist asymmetrisch, d.h. wenn ein Huhn ein Anderes pickt, dann traut sich dieses auf keinen Fall zurück zu picken. Weiteres pickt für *jedes* Paar an Hühnern entweder das Eine das Andere, oder umgekehrt. Es geht also ziemlich zu auf Florians Hof.

Interessanterweise ist ihm aufgefallen, dass dieses Hühnerpicken nicht zwingend transitiv ist. Das heißt, dass es z.B. ein Trio A, B, C an Hühnern geben kann, sodass $A \rightarrow B$ pickt, $B \rightarrow C$ pick - aber den noch $C \rightarrow A$ pickt. Es herrscht also keine besondere Rangordnung zwischen den Hühnern.

Jeden Abend treibt Florian seine Hühner wieder zurück in den Stall. Dazu versucht er diese in einer Reihe anzuordnen, sodass sie dann, eines nach dem anderen, hinein gehen können. Besorgt um die Gesundheit seiner Tiere möchte Florian, dass in dieser Reihe möglichst selten ein Huhn seinen Vorgänger pickt.

Nachdem er aber etwas aus der Übung im Competitive Programming ist, ist es nun deine Aufgabe dieses Problem zu lösen. Finde eine Permutation, sodass - wenn man die Hühner in dieser Reihenfolge aufstellt - möglichst selten ein Huhn seinen Vorgänger pickt. Gibt es mehrere optimale Anordnungen, so kannst du irgendeine davon nehmen.

Eingabe

Die erste Zeile des Inputs enthält n , die Anzahl an Hühner. Die folgenden n Zeilen beschreiben die Pick-Relationen. Die i -te davon gibt an, welche Hühner das i -te Huhn picken. Eine 1 an der j -ten Stelle dieser Zeile bedeutet, dass das i -te Huhn das j -te pickt - und eine 0, dass es das j -te Huhn nicht pickt.

Ausgabe

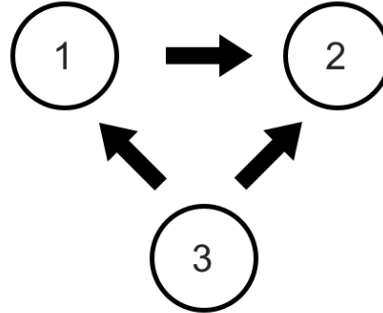
Gib eine Permutation von $1, 2, \dots, n$ aus, sodass - wenn man die Hühner in dieser Reihenfolge aufstellt - möglichst selten ein Huhn seinen Vorgänger pickt. Gibt es mehrere optimale Anordnungen, so kannst du irgendeine davon ausgeben.

Beispiel

Eingabe	Ausgabe
3 0 1 0 0 0 0 1 1 0	2 1 3

Florian hat folgende 6 Möglichkeiten seine Hühner in den Stall zu treiben. Die Zahl daneben gibt an, wie oft bei dieser Möglichkeit ein Huhn seinen Vorgänger pickt:

- 1 2 3 1
- 1 3 2 1
- 2 1 3 0
- 2 3 1 1
- 3 1 2 2
- 3 2 1 1



Im Fall 1 2 3 pickt Huhn 1 seinen Vorderen (2), Huhn 2 lässt aber Huhn 3 in Ruhe. Somit ist hier die Anzahl an Picks gleich 1. Stell Florian die Hühner in der Reihenfolge 2 1 3 auf, so wird kein Huhn von seinem Vorgänger belästigt. Somit ist in diesem Fall die (einzig richtige) Antwort 2 1 3.

Subtasks

Allgemein gilt:

- $1 \leq n \leq 1000$
- Das Picken ist asymmetrisch (wie oben erklärt)
- Für jedes Paar an (verschiedenen) Hühnern pickt entweder das Eine das Andere, oder umgekehrt

Subtask 1 (13 Punkte): $1 \leq n \leq 10$

Subtask 2 (18 Punkte): $1 \leq n \leq 16$

Subtask 3 (21 Punkte): $1 \leq n \leq 100$

Subtask 4 (48 Punkte): Keine Einschränkungen

Beschränkungen

Zeitlimit: 1 s **Speicherlimit:** 256 MB