Computer Network

There are N computers in the computer classroom of a local secondary school and they are connected with cables to form a single network. Each cable joins two distinct computers. Some pairs of computers may not be joined by a cable directly, but a message can always be sent from any computer to any other computer through intermediate computers joined by cables. A message always chooses the shortest route to travel: the number of *intermediate* computers along it's path (i.e. computers other than the sender and the receiver that the message visits) is minimised.

Adam and Billy, who use distinct computers a and b in this classroom, wish to determine a shortest route between their computers. They do not know the layout of the cables but they can send messages between all pairs of computers and calculate the number of intermediate computers that they visit.

However, Adam and Billy are not very good with computers and they ask you for your help to achieve their goal without sending too many messages.

Task

Find a shortest route between computers a and b without sending more messages than is allowed.

Implementation

You need to implement one procedure `findRoute(N, a, b)` that takes the following parameters:

- N — the number of computers in the classroom (they are labelled from 1 to N)
- a, b — the labels of Adam and Billy's computers ($a \neq b$ and $1 \leq a, b \leq N$)

Your procedure `findRoute` can call function `ping(i, j)` that takes as parameters two distinct labels of computers ($i \neq j$ and $1 \leq i, j \leq N$) and returns the number of intermediate computers that a message travelling from i to j would visit.

Your procedure `findRoute` has to describe a shortest route that a message sent from a to b might take. This must be done by repeatedly calling procedure `travelTo(k)` that takes as parameter the label of computer to which the message should travel next ($1 \leq k \leq N$). The message starts at a, and whenever `travelTo(k)` is called, it moves to computer k.

In addition to the standard requirements (time and memory limits, no runtime errors), your submission has to achieve the following in order to solve a testcase:

- the message must be at b when procedure `findRoute` terminates,
- any two consecutive computers in its path must be joined by a cable,
- it must take a shortest route,

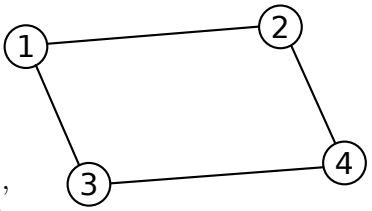
- the number M of calls to function `ping` must not exceed the allowed limit (see section Scoring),
- function `ping` and procedure `travelTo` must only be called with allowed parameter values.

Example

Consider the example shown in the diagram below (points correspond to computers, lines — to cables). There are $N = 4$ computers in total, and Adam and Billy are using computers $a = 1$ and $b = 4$.

The first call will be to procedure

```
findRoute(4, 1, 4).
```



Its sample behaviour could be as follows:

`ping(1, 4)` is called and returns 1,
`ping(1, 2)` is called and returns 0,
`ping(2, 4)` is called and returns 0.

This information is sufficient to determine that $1 \to 2 \to 4$ is a shortest route from 1 to 4. This should be described as follows:

`travelTo(2)` is called,
`travelTo(4)` is called,
`findRoute` terminates.

Scoring

In all subtasks the constraint $2 \leq N \leq 1000$ holds.

Subtask 1 (25 points): between any two computers there is exactly one shortest route; M cannot exceed $2N$.

Subtask 2 (25 points): M cannot exceed N^2.

Subtask 3 (25 points): M cannot exceed $4N$.

Subtask 4 (25 points): M cannot excced $2N$.

Constraints

Time limit: 1 s

Memory limit: 64 MB

Experimentation

The sample grader on your computer will read data from standard input. The first line of input should contain four integers N, a, b and the allowed limit for M. The next N lines should contain N integers each, describing the layout of the cables: the j–th integer in the i–th of these rows ($i \neq j$) should be the number of intermediate computers that a message sent from i to j would visit. It does not matter what the entries with $i = j$ are.

The following input describes the above example with M limited by 100:

```
4 1 4 100
0 0 0 1
0 0 1 0
0 1 0 0
1 0 0 0
```