

“Imperial Messengers”

Problem The empire has a number of cities. For communicating important messages from the capitol to other cities, a network of messengers is going to be set up. Some number of messengers will be stationed in the capitol city. Each messenger will ride to one other city, where the message will be posted in the town square, and handed off to some number of new messengers, each of whom will ride off to a different city, repeating the process. The empire’s unemployment rate is such that there are no limits placed upon the number of messengers in each city, the only goal is to get the message communicated throughout the empire as quickly as possible.

Input The input will describe the routes between the n cities. All cities are reachable using some path from the capitol city. The first line of the input will be n , the number of cities, such that $1 \leq n \leq 100$. The rest of the input defines an adjacency matrix, A . The adjacency matrix is square and of size $n \times n$. Each of its entries will be either an integer or the character x . The value of $A_{(i,j)}$ indicates the time required to travel from city i to city j . A value of x for $A_{(i,j)}$ indicates that a message cannot be sent directly from city i to city j . Note that for a city to send a message to itself does not require a messenger, so $A_{(i,i)} = 0$ for $1 \leq i \leq n$. Also, you may assume that the adjacency matrix is undirected (messengers can travel in either direction in equal time), so that $A_{(i,j)} = A_{(j,i)}$. Thus only the entries on the (strictly) lower triangular portion of A will be supplied as input. The input to your program will be the lower triangular section of A . That is, the second line of input will contain one entry, $A_{(2,1)}$. The next line will contain two entries, $A_{(3,1)}$ and $A_{(3,2)}$, and so on.

Output Your program should output the minimum time required before a message sent from the capitol (city #1) is known throughout the empire, i.e. the time it is received in the last city to get the message.

Sample Input

```
5
50
30 5
100 20 50
10 x x 10
```

Output for the Sample Input

```
35
```