The Gold-diggers

Time limit: *1000 ms*
Memory limit: *256 MB*

Given a simple connected graph with vertices indexed from 0 to $N-1$, N being the order of the graph, two players start at vertex 0. We call the players **IEEE** and **Xtreme**.

There are 7 gold coins, located at vertices $N-7, N-6, \ldots, N-1$ (one per vertex).

The aim of the game is to catch as many gold coins as possible.

We call **move** for a player the change in location from the vertex he is currently at to one of its neighbors (i.e. if the player is at vertex u, it can go to any vertex v such that (u, v) is an edge in the graph).

The game is played as follows: alternatively and starting with **IEEE**, each player makes a **move**. If the vertex the player goes to contains a gold coin, it is immediately added to its collection and removed from the vertex. The two players can be at the same location at any time (and they start at the same vertex), but if that vertex initially contained a gold coin, only the first player to make the **move** collects it.

We call **strategy** for **IEEE** (respectively for **Xtreme**) a function that takes as input the current state of the game (position of the players and remaining gold coins) as well as the history of all **moves**, and associates a **move** for **IEEE** (respectively for **Xtreme**).

Thus, given one strategy for **IEEE** and one for **Xtreme** it is possible to compute how many gold coins are taken by **IEEE** during the game. Note that the game may be infinite.

We call **value** of a strategy for **IEEE** the **minimum** (i.e. provided that both players have acted in a way that maximized the gold coin collection, knowing that the strategy of the other player would similarly be to maximize the acquisition of gold coins) number of gold coins **IEEE** takes, considering any possible strategy for **Xtreme**.

We would like to know what is the **value** of the best strategy (the one with the maximum value) for **IEEE**.

Standard input

The input of your program is organized as follows:

- The first line contains only the order of the graph, N.
- The $i+1^{\text{th}}$ line contains the list of neighbors of vertex i, separated by spaces, in no specific order.

Standard output

Print the answer on the first line.

Constraints and notes

- $8 \leq N \leq 20$
- (u, v) is an edge if and only if (v, u) is an edge for all valid u and v

Input	Output	Explanation

`10`
`1`
`0 2`
`1 3`
`2 4`
`3 5`
`4 6`
`5 7`
`6 8`
`7 9`
`8`
	`7`	This graph is in fact a straight line with the 10 vertices positioned linearly on it. Since the number of vertices is $N = 10$, the gold coins would be positioned on vertices $3, 4, 5, 6, 7, 8$ and 9. As there is only a single way to go from vertex u to the vertex $u + 1$ (e.g. from 3 to 4) and the **IEEE** player always makes the first move, he would always come to each of the gold-containing vertices first (followed by the **XTREME** player on the next move) and hence the **minimum** number of gold coins the **IEEE** player would acquire would be 7. The exact game play would be as follows:

#	Player	Move
1	IEEE	$0 \to 1$
2	XTREME	$0 \to 1$
3	IEEE	$1 \to 2$
4	XTREME	$1 \to 2$
5	IEEE	$2 \to 3$ (gets 1 coin)
6	XTREME	$2 \to 3$
7	IEEE	$3 \to 4$ (gets 1 coin)
8	XTREME	$3 \to 4$
9	IEEE	$4 \to 5$ (gets 1 coin)
10	XTREME	$4 \to 5$
11	IEEE	$5 \to 6$ (gets 1 coin)
12	XTREME	$5 \to 6$
13	IEEE	$6 \to 7$ (gets 1 coin)
14	XTREME	$6 \to 7$

Input	Output	Explanation

#	Player	Move
15	IEEE	$7 \to 8$ (gets 1 coin)
16	XTREME	$7 \to 8$
17	IEEE	$8 \to 9$ (gets 1 coin)
18	XTREME	$8 \to 9$

Input:

```
10
1 2 3 4 5 6 7 8 9
0 2 3 4 5 6 7 8 9
0 1 3 4 5 6 7 8 9
0 1 2 4 5 6 7 8 9
0 1 2 3 5 6 7 8 9
0 1 2 3 4 6 7 8 9
0 1 2 3 4 5 7 8 9
0 1 2 3 4 5 6 8 9
0 1 2 3 4 5 6 7 9
0 1 2 3 4 5 6 7 8
```

Output:

```
4
```

This graph is a complete graph with 10 vertices. As such, it is possible to move from any vertex to any vertex.

There are two cases: either **IEEE** decides to move each turn to a vertex that contains a gold coin, or at some point it goes to a vertex that does not contain a gold coin. In both cases, **XTREME** can decide to move systematically to a vertex that contains a gold coin. If the first case, and since **IEEE** plays first, he will catch exactly 4 gold coins. In the second case, he will catch at most 3 gold coins. We conclude that the value of the best strategy for **IEEE** is exactly 4.

In more details, a tight play would be:

#	Player	Move
1	IEEE	$0 \to 3$ (gets 1 coin)
2	XTREME	$0 \to 4$ (gets 1 coin)
3	IEEE	$3 \to 5$ (gets 1 coin)
4	XTREME	$4 \to 6$ (gets 1 coin)
5	IEEE	$5 \to 7$ (gets 1 coin)
6	XTREME	$6 \to 8$ gets (1 coin)
7	IEEE	$7 \to 9$ (gets 1 coin)

Other tight plays could be obtained by renaming vertices 3 to 9