

Problems

Problem	COMFORT	PEOPLE	TEAMS
Source file	comfort.pas comfort.c comfort.cpp	people.pas people.c people.cpp	teams.pas teams.c teams.cpp
Input file	comfort.in	people.in	teams.in
Output file	comfort.out	people.out	teams.out
Time limit per test	10 seconds	10 seconds	10 seconds
Number of tests	10	10	10
Points per test	3	5	7
Total points	30	50	70
	150		

COMFORT

A game-board consists of N fields placed around a circle. Fields are successively numbered from 1 to N clockwise. On **some** of the fields there are obstacles.

Player starts on a field marked with number **1**. His goal is to reach a given field marked with number Z using only moves consisting of clockwise jumps of **length K** . The player's path should not contain **any** of the fields having an obstacle.

For example, if $N=13$, $K=3$ and $Z=9$, the player can jump across the fields 1, 4, 7, 10, 13, 3, 6 and 9, reaching his goal under condition that none of these fields has an obstacle on it.

Your task is to write a program that finds the **smallest** possible number K .

Input data

First line of the input file consists of integers N , Z and M , $2 \leq N \leq 1000$, $2 \leq Z \leq N$, $0 \leq M \leq N-2$.

N represents number of fields on the game-board and Z is a given goal-field.

Next line consists of M different integers that represent marks of fields having an obstacle. Fields marked 1 and Z **do not contain** an obstacle.

Output data

The first and only line of the output file should contain requested number K defined above.

Examples

`comfort.in`

```
7 4 1
6
```

`comfort.out`

```
1
```

`comfort.in`

```
9 7 2
2 3
```

`comfort.out`

```
3
```

`comfort.in`

```
7 6 2
2 4
```

`comfort.out`

```
5
```

PEOPLE

A group of N people is given. These people can be classified into **two subgroups**: those who always tell the **truth** and those who **lie**.

We interviewed all of them. Each person named several members of the group and said whether they belong to the first subgroup or to the second.

Your task is to write a program that will **use these statements to determine** who tells a lie and who doesn't.

Input data

First line of the input file consists of an integer N , $2 \leq N \leq 20$, a number of people in the group.

Next $2N$ lines contain statements of each person in the following form (j -th person's statement is placed in lines $2j$ and $2j+1$):

K A_1 A_2 ... A_K

L B_1 B_2 ... B_L

This means that j -th person said that A_1, A_2, \dots, A_K tell the truth and that B_1, B_2, \dots, B_L lie.

People are numbered 1 to N . Each person made a statement about at least one other, did not make any statements about herself and did not say that someone tells the truth and lies at the same time.

If a person tells the truth then **each** of her statements must be **true** and if she lies then **at least one** of her statements must be **false**.

Output data

The output file should have N lines, j -th line telling whether j -th person tells the truth or lies. If a person tells the truth, word 'true' should be printed and word 'false' otherwise.

Test data will be such that the solution will always exist, although it may not be necessarily unique.

Examples

people.in

```
3
1 3
0
0
1 1
1 1
0
```

people.out

```
true
false
true
```

people.in

```
4
1 2
1 3
1 4
0
0
1 4
0
1 3
```

people.out

```
true
true
false
true
```

people.in

```
4
1 2
0
0
1 4
0
1 4
2 1 3
1 2
```

people.out

```
true
true
true
false
```

TEAMS

Friends from the neighbourhood gathered on the school playground to play a football match. They need to be divided into two teams. Players have T-shirts numbered 1 to N in a way that **the best player** has number **1** and **the worst player** has number **N**.

Teams must have an **equal** number of players and are formed in the following way. Each of the players makes a list of his colleagues with whom he **doesn't want** to be in the same team. Naturally, everyone wants to play with those who are **better** players than him and therefore his list **contains only players worse** than he is.

Your task is to write a program that will separate friends into two teams having an equal number of players in order that **each** player's wishes are respected.

Input data

First line of the input file contains of an **even** integer N, $2 \leq N \leq 1000$, a number of friends.

Each of the next N lines has a single player's wish-list, (i+1)-th line having an i-th player's wish-list in the following form:

$K \ A_1 \ A_2 \ \dots \ A_K$

This means that i-th player doesn't want any of A_1, A_2, \dots, A_K in his team.

Output data

The first and only line of the output file should contain T-shirt numbers of players of either of the teams, separated by a single space. Ordering of the players is irrelevant.

Test data will be such that the solution will always exist, although it may not be necessarily unique.

Examples

teams.in

```
4
1 4
1 3
1 4
0
```

teams.out

```
1 3
```

teams.in

```
6
2 3 5
1 4
0
0
1 6
0
```

teams.out

```
6 4 1
```

teams.in

```
8
1 5
0
1 7
0
0
0
0
0
```

teams.out

```
1 6 8 7
```