

Codeforces Round #503 (Div. 2)

Problem B – Badge

Prof. Edson Alves

Faculdade UnB Gama

In Summer Informatics School, if a student doesn't behave well, teachers make a hole in his badge. And today one of the teachers caught a group of n students doing yet another trick.

Let's assume that all these students are numbered from 1 to n . The teacher came to student a and put a hole in his badge. The student, however, claimed that the main culprit is some other student p_a .

After that, the teacher came to student p_a and made a hole in his badge as well. The student in reply said that the main culprit was student p_{p_a} .

Na Escola de Verão de Informática, se um estudante não se comporta bem, os professores fazem um furo em sua insígnia. Hoje um dos professores flagrou um grupo de n estudantes fazendo outra travessura.

Assuma que os estudantes estão numerados de 1 a n . O professor foi ao estudante a e furou sua insígnia. O estudante, porém, afirmou que o principal culpado é o estudante p_a .

Após isso, o professor foi ao estudante p_a e furou sua insígnia também. Em resposta o estudante disse que o culpado foi o estudante p_{p_a} .

This process went on for a while, but, since the number of students was finite, eventually the teacher came to the student, who already had a hole in his badge.

After that, the teacher put a second hole in the student's badge and decided that he is done with this process, and went to the sauna.

You don't know the first student who was caught by the teacher. However, you know all the numbers p_i . Your task is to find out for every student a , who would be the student with two holes in the badge if the first caught student was a .

Este processo continuou mas, como o número de estudantes é finito, em algum momento o professor chegou em um estudante que já tinha um furo em sua insígnia.

Após isso, o professor fez um segundo furo na insígnia do estudante, decidiu que o processo estava encerrado e foi para a sauna.

Você não sabe quem foi o primeiro estudante que foi pego pelo professor. Contudo, você conhece todos os valores p_i . Sua tarefa é determinar, para cada estudante a , quem será o estudante com dois furos na insígnia se o primeiro a ser pego foi a .

Input

The first line of the input contains the only integer n ($1 \leq n \leq 1000$) – the number of the naughty students.

The second line contains n integers p_1, \dots, p_n ($1 \leq p_i \leq n$), where p_i indicates the student who was reported to the teacher by student i .

Output

For every student a from 1 to n print which student would receive two holes in the badge, if a was the first student caught by the teacher.

Entrada

A primeira linha da entrada contém um único inteiro n ($1 \leq n \leq 1000$) – o número de estudantes malcriados.

A segunda linha contém n inteiros p_1, \dots, p_n ($1 \leq p_i \leq n$), onde p_i indica o estudante que foi acusado ao professor pelo estudante i .

Saída

Para cada estudante a de 1 a n imprima qual estudante irá receber dois furos em sua insígnia, se a foi o primeiro estudante capturado pelo professor.

Exemplo de entrada e saída

Exemplo de entrada e saída

3

Exemplo de entrada e saída

3



de estudantes

Exemplo de entrada e saída

3

2 3 2

Exemplo de entrada e saída

3

2 3 2

↑ ↑ ↑

p_1 p_2 p_3

Exemplo de entrada e saída

3

2 3 2

1

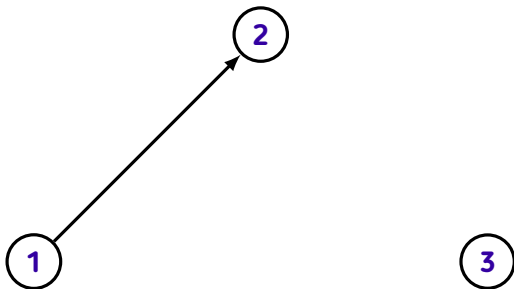
2

3

Exemplo de entrada e saída

3

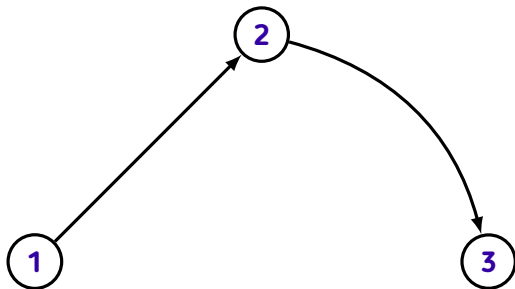
2 3 2



Exemplo de entrada e saída

3

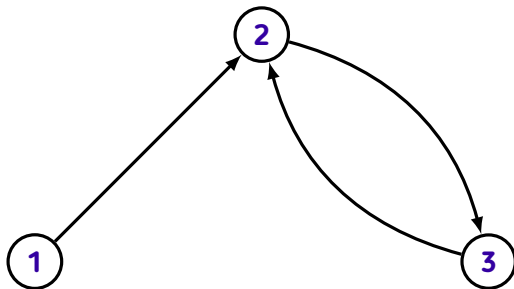
2 3 2



Exemplo de entrada e saída

3

2 3 2



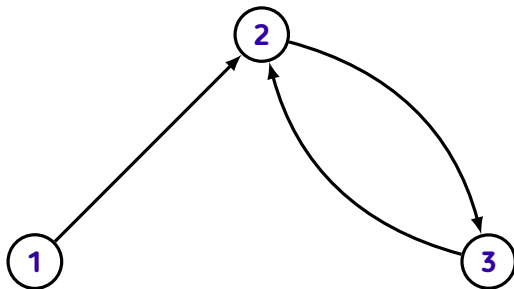
Exemplo de entrada e saída

3

2 3 2



2



Exemplo de entrada e saída

3

2 3 2



2

2

1

3

Exemplo de entrada e saída

3

2 3 2



2



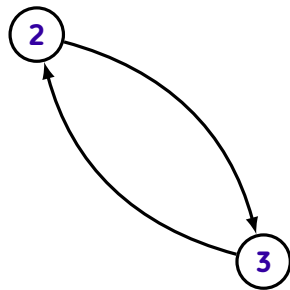
Exemplo de entrada e saída

3

2 3 2



2



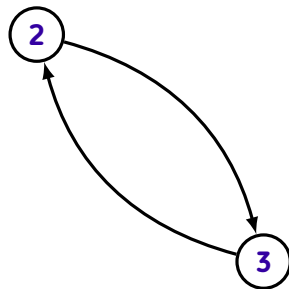
Exemplo de entrada e saída

3

2 3 2



2 2



Exemplo de entrada e saída

3

2 3 2



2 2

1

2

3

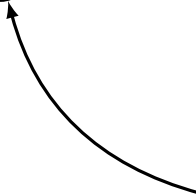
Exemplo de entrada e saída

3

2 3 2



2 2



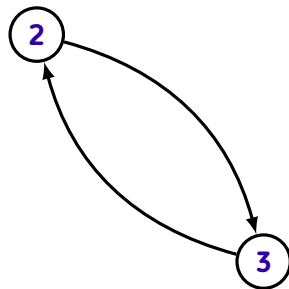
Exemplo de entrada e saída

3

2 3 2



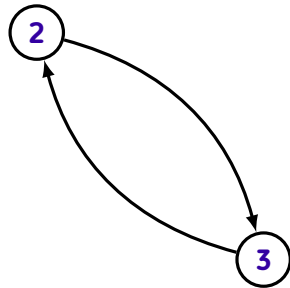
2 2



Exemplo de entrada e saída

3

2 3 2
↓ ↓ ↓
2 2 3



Exemplo de entrada e saída

Exemplo de entrada e saída

3

1 2 3

Exemplo de entrada e saída

3

1 2 3

1

2

3

Exemplo de entrada e saída

3

1 2 3



Exemplo de entrada e saída

3

1	2	3
↓	↓	↓
1	2	3



Solução

Solução

- ★ Para cada estudante a são necessárias, no máximo, $n + 1$ verificações

Solução

- ★ Para cada estudante a são necessárias, no máximo, $n + 1$ verificações
- ★ Estas verificações podem ser feitas por meio de uma DFS

Solução

- ★ Para cada estudante a são necessárias, no máximo, $n + 1$ verificações
- ★ Estas verificações podem ser feitas por meio de uma DFS
- ★ Há n estudantes distintos

Solução

- ★ Para cada estudante a são necessárias, no máximo, $n + 1$ verificações
- ★ Estas verificações podem ser feitas por meio de uma DFS
- ★ Há n estudantes distintos
- ★ A complexidade da solução é $O(n^2)$

```
#include <bits/stdc++.h>

using namespace std;

const int MAX { 1010 };

bitset<MAX> found;

int dfs(int u, const vector<int>& ps)
{
    if (found[u])
        return u;

    found[u] = true;

    return dfs(ps[u], ps);
}
```

```
vector<int> solve(int N, const vector<int>& ps)
{
    vector<int> ans(N + 1);

    for (int u = 1; u <= N; ++u)
    {
        found.reset();
        ans[u] = dfs(u, ps);
    }

    return ans;
}
```