Problem 3 – Indices

You are given a zero-based array **ARR** with **N** integer numbers in it. Each element of **ARR** is an index in the **ARR** (seems like a recursive definition, right?).

You are also given the sequence that starts from the first element (**0**) then moves to the element with index **ARR[0]**, then moves to the element with index **ARR[ARR[0]]**, then moves to the element with index **ARR[ARR[ARR[0]]],** and so on…

The full sequence is generated by performing these actions until you reach an index that is outside the bounds of the array **ARR**. Of course cycles are absolutely possible. When a cycle is started in the sequence it may never reach any index that is outside the bounds of the **ARR**.

Write a program that outputs the elements in the given sequence. When you find cycle you should output it in round brackets as shown in the examples below. Please note that no spaces should be printed between the brackets and the number.

Input

The input data should be read from the console.

In the first input line you are given the number **N** of the elements in **ARR**.

From the second line you should read the numbers in the array **ARR** separated by a single space.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data should be printed on the console.

On the only output line you should print the described sequence. All the cycles should be printed with round brackets, with no spaces between the brackets and the numbers.

Constraints

- **N** will be between 1 and 200 000, inclusive.
- Numbers in ARR will be between -2 000 000 000 and 2 000 000 000, inclusive.
- Allowed working time for your program: 0.25 seconds.
- Allowed memory: 16 MB.

Examples

Input example	Output example
6 1 2 3 5 7 8	0 1 2 3 5
6 1 2 3 5 7 1	0(1 2 3 5)