A. Assembling Pokemon Team

time limit per test: 2 s.
memory limit per test: 256 MB

Ah, you must be the new trainer! Welcome to Professor Oak's Lab, where we study the how to form Pokémon teams. I have an important task for you—one that only the sharpest minds can solve.

I have a list of Pokémon levels, and your mission is to extract the longest possible subsequence that forms an increasing chain of consecutive levels. Think of it like assembling a perfect battle-ready team!

In more precise terms, you must find the longest "subsequence" that follows this value:

$$[x, x+1, \ldots, x+k-1]$$

for some starting level (x) and length (k).

But remember—just like Pokémon battles, order matters! You can remove pokenmons, but you must keep the original order of the remaining ones.

Input

The first line of the input contains an integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of Pokémon levels. The second line contains n integers $x_1, x_2, \ldots, x_n (1 \le x_i \le 10^9)$ — the Pokémon levels.

Output

On the first line, print (k) — the maximum length of the subsequence of the given array that forms an increasing sequence of consecutive levels.

On the second line, print the sequence of the indices of the **any** maximum length subsequence of the given array that forms an increasing sequence of consecutive levels.

Examples

input Copy

```
7
3 3 4 7 5 6 8
```

output Copy

```
4
2 3 5 6
```

input Copy

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

output Copy

```
2
1 4
```

input Copy

```
4
10 9 8 7
```

output Copy

```
1
1
```

input Copy

```
9
6 7 8 3 4 5 9 10 11
```

output Copy

```
6
1 2 3 7 8 9
```

Note

All valid answers for the first example (as sequences of indices):

- $[1, 3, 5, 6]$
- $[2, 3, 5, 6]$

All valid answers for the second example:

- $[1, 4]$
- $[2, 5]$
- $[3, 6]$

All valid answers for the third example:

- $[1]$
- $[2]$
- $[3]$
- $[4]$

All valid answers for the fourth example:

- $[1, 2, 3, 7, 8, 9]$

- DP Knapsack - Preclass
- DP Edit Distance - Preclass
- Segment Tree - Homework
- DP Palindromes - Preclass
- Lazy Segment Tree - Homework
- LCA and Binary Lifting - Homework
- DP intro - Preclass
- Square Root Decomposition - Homework
- DP Longest Increasing Subsequence - Preclass
- Greedy - Preclass
- Fenwick Tree - Preclass
- Bit Manipulation - Homework
- Square Root Decomposition - Preclass
- Fast Exponentiation - Homework
- MST - Homework
- Lazy Segment Tree - Preclass
- LCA and Binary Lifting - Preclass
- Segment Tree - Preclass
- Bit Manipulation - Preclass
- Fast Exponentiation - Preclass
- MST - Preclass
- Graph Traversal 2 - Homework
- Graph Traversal 2 - In Class
- All Pairs Shortest Path - Homework
- All Pairs Shortest Path - In Class
- Single Source Shortest Path - Homework
- Single Source Shortest Path - In Class
- Graph Traversal 1 - Homework
- Graph Traversal 1 - In Class
- Binary Search Tree - Homework
- Binary Search Tree - In Class
- Disjoint Sets - Homework
- Disjoint Sets - In Class
- Divide and Conquer - Homework
- Divide and Conquer - In Class
- Complete Search - Homework
- Complete Search - In Class
- STL - Homework
- STL - In Class
- IO Problems - Preclass
- Test Contest