G. Good Array

time limit per test: 1 second
memory limit per test: 256 megabytes

Let's call an array *good* if there is an element in the array that equals to the sum of all other elements. For example, the array $a = [1, 3, 3, 7]$ is good because there is the element $a_4 = 7$ which equals to the sum $1 + 3 + 3$.

You are given an array a consisting of n integers. Your task is to print all indices j of this array such that after removing the j-th element from the array it will be *good* (let's call such indices *nice*).

For example, if $a = [8, 3, 5, 2]$, the *nice* indices are 1 and 4:

- if you remove a_1, the array will look like $[3, 5, 2]$ and it is *good*;
- if you remove a_4, the array will look like $[8, 3, 5]$ and it is *good*.

You have to consider all removals **independently**, i. e. remove the element, check if the resulting array is *good*, and return the element into the array.

Input

The first line of the input contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the number of elements in the array a.

The second line of the input contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^6$) — elements of the array a.

Output

In the first line print one integer k — the number of indices j of the array a such that after removing the j-th element from the array it will be *good* (i.e. print the number of the *nice* indices).

In the second line print k distinct integers $j_1, j_2, \dots, j_k$ in **any** order — *nice* indices of the array a.

If there are no such indices in the array a, just print 0 in the first line and leave the second line empty or do not print it at all.

Examples

input Copy

```
5
2 5 1 2 2
```

output Copy

```
3
4 1 5
```

input Copy

```
4
8 3 5 2
```

output Copy

```
2
1 4
```

input Copy

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

output Copy

```
0
```

Note

In the first example you can remove any element with the value 2 so the array will look like $[5, 1, 2, 2]$. The sum of this array is 10 and there is an element equals to the sum of remaining elements ($5 = 1 + 2 + 2$).

In the second example you can remove 8 so the array will look like $[3, 5, 2]$. The sum of this array is 10 and there is an element equals to the sum of remaining elements ($5 = 3 + 2$). You can also remove 2 so the array will look like $[8, 3, 5]$. The sum of this array is 16 and there is an element equals to the sum of remaining elements ($8 = 3 + 5$).

In the third example you cannot make the given array *good* by removing exactly one element.

ICPC Assiut University Training - Juniors Phase 1 Sheets-2022

Public

Participant

☆

→ Group Contests

- Juniors Phase 1 Practice #5 (Bitmask, Bitset, Bits)
- Juniors Phase 1 Practice #4 (Binary search , Two pointers)
- Juniors Phase 1 Practice #3 (STL 2)
- Juniors Phase 1 Practice #2 (STL 1)
- Juniors Phase 1 Practice #1 (Prefix sum , Frequency Array)

Juniors Phase 1 Practice #1 (Prefix sum , Frequency Array)

Finished

Practice

☆

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Submit?

Language: GNU G++20 13.2 (64 bit, win

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
228030518	Oct/13/2023 16:33	Accepted
228011760	Oct/13/2023 13:55	Accepted
228011452	Oct/13/2023 13:51	Runtime error on test 6