Code Jam 2022 - Round 1A

Equal Sum

Problem

You are given a set of distinct integers. You need to separate them into two non-empty subsets such that each element belongs to exactly one of them and the sum of all elements of each subset is the same.

An anonymous tip told us that the problem above was unlikely to be solved in polynomial time (or something like that), so we decided to change it. Now **you** get to decide what half of the integers are!

This is an interactive problem with three phases. In phase 1, you choose $\mathbf{N}$ distinct integers. In phase 2, you are given another $\mathbf{N}$ integers that are distinct from each other and from the ones you chose in phase 1. In phase 3, you have to partition those $2\mathbf{N}$ integers into two subsets, both of which sum to the same amount. All $2\mathbf{N}$ integers are to be between 1 and 10^9, inclusive, and it is guaranteed that they sum up to an even number.

Input and output

This is an interactive problem. You should make sure you have read the information in the Interactive Problems section of our FAQ.

Initially, your program should read a single line containing an integer, $\mathbf{T}$, the number of test cases. Then, $\mathbf{T}$ test cases must be processed.

For each test case, your program must first read a line containing a single integer $\mathbf{N}$. Then, it must output a line containing $\mathbf{N}$ distinct integers $A_1, A_2, \ldots, A_\mathbf{N}$. Each of these integers must be between 1 and 10^9, inclusive. After that, your program must read a line containing $\mathbf{N}$ additional integers $\mathbf{B_1}, \mathbf{B_2}, \ldots, \mathbf{B_N}$. Finally, your program must output a line containing between 1 and $2\mathbf{N} - 1$ integers from among $A_1, A_2, \ldots, A_\mathbf{N}, \mathbf{B_1}, \mathbf{B_2}, \ldots, \mathbf{B_N}$: the ones chosen to be part of the first subset. The integers from A and $\mathbf{B}$ that you do not output are considered to be part of the other subset.

The next test case starts immediately if there is one. If this was the last test case, the judge will expect no more output and will send no further input to your program. In addition, all $\mathbf{T}$ test cases are always processed, regardless of whether the final output from your program is correct or not.

Note: It can be shown that given the limits for this problem, there exists a sequence $A_1, A_2, \ldots, A_\mathbf{N}$ such that any sequence $\mathbf{B_1}, \mathbf{B_2}, \ldots, \mathbf{B_N}$ results in a set of $2\mathbf{N}$ integers that can be separated into two subsets with equal sums.

If the judge receives an invalidly formatted or invalid line (like outputting an unexpected number of integers, or integers out of range, or repeated integers in a line) from your program at any moment, the judge will print a single number -1 and will not print any further output. If your program continues to wait for the judge after receiving a -1, your program will time out, resulting in a Time Limit Exceeded error. Notice that it is your responsibility to have your program exit in time to receive a Wrong Answer judgment instead of a Time Limit Exceeded error. As usual, if the memory limit is exceeded, or your program gets a runtime error, you will receive the appropriate judgment.

Limits

Time limit: 5 seconds.
Memory limit: 1 GB.

Test Set 1 (Visible Verdict)

$1 \le \mathbf{T} \le 100$.
$\mathbf{N} = 100$.
$1 \le \mathbf{B_i} \le 10^9$, for all i.
$\mathbf{B_i} \ne A_j$, for all i, j.
$\mathbf{B_i} \ne \mathbf{B_j}$, for all $i \ne j$.
For each test case, the judge will choose the $\mathbf{B_i}$s such that the sum of all $2\mathbf{N}$ integers is even.

Testing Tool

You can use this testing tool to test locally or on our platform. To test locally, you will need to run the tool in parallel with your code; you can use our interactive runner for that. For more information, read the instructions in comments in that file, and also check out the Interactive Problems section of the FAQ.

Instructions for the testing tool are included in comments within the tool. We encourage you to add your own test cases. Please be advised that although the testing tool is intended to simulate the judging system, it is **NOT** the real judging system and might behave differently. If your code passes the testing tool but fails the real judge, please check the Coding section of the FAQ to make sure that you are using the same compiler as us.

Download testing tool

Sample Interaction

Judge **Solution**

Number of cases

`2`

Case 1

`3`

Judge gives $\mathbf{N} = 3$.

`5 1 3`

Solution starts with $\{1, 3, 5\}$.

`10 4 9`

Judge adds to the set to get $\{1, 3, 4, 5, 9, 10\}$.

`1 10 5`

Solution says to use $\{1, 5, 10\}$ *for one subset.*
This is correct because $1 + 5 + 10 = 16 = 3 + 4 + 9$.

Case 2

`3`

Judge gives $\mathbf{N} = 3$.

```
5 2 3
```

Solution starts with $\{2, 3, 5\}$.

```
10 8 12
```

Judge adds to the set to get $\{2, 3, 5, 8, 10, 12\}$.

```
12 8
```

Solution says to use $\{8, 12\}$ *for one subset.*
This is correct because $8 + 12 = 20 = 2 + 3 + 5 + 10$.

In the sample interaction above, the solution gets all cases right and would receive a correct verdict. Notice that the value for $\mathbf{N}$ does not conform to the limits for the Test Set and is used to simplify the example only. Notice that the judge could have given the solution the integers $\{2, 7, 100\}$ for the first case, making it impossible for the solution to find a valid partition into subsets of equal sum.