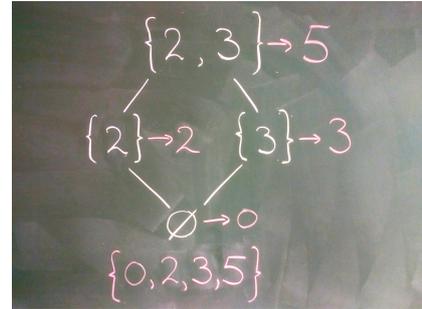


K Knapsack Packing

One of the most difficult things about going on a holiday is making sure your luggage does not exceed the maximum weight. You, chairman of the Backpacker's Association for Packing Carry-ons, are faced with exactly this problem. You are going on a lovely holiday with one of your friends, but now most of your time is spent in frustration while trying to pack your backpack. In order to optimize this process, you and your friend have independently set upon trying to find better ways to pack.



After some time you have a serious breakthrough! Somehow you managed to solve the Knapsack problem in polynomial time, defying expectations everywhere. You are not interested in any theoretical applications, so you immediately return to your friend's apartment in order to now quickly pack your backpack optimally.

When you arrive there, you find that your friend has set upon her own solution, namely to enumerate all possible packings. This means that all items you possibly wanted to bring are scattered across the entire apartment, and it would take a really long time to get all the items back together.

Luckily you can use the work your friend has done. For every possible subset of items that you can possibly bring, she has written down the total weight of these items. Alas, she did not write down what items were part of this total, so you do not know what items contributed to each total weight. If the original weights of the items formed a collection (a_1, \dots, a_n) of non-negative integers, then your friend has written down the multiset

$$S((a_1, \dots, a_n)) := \left\{ \sum_{i \in I} a_i \mid I \subseteq \{1, \dots, n\} \right\}.$$

For example, if your friend had two items, and the weights of those two items are 2, 3, then your friend has written down

- 0, corresponding to the empty set $\{\}$;
- 2, corresponding to the subset $\{2\}$;
- 3, corresponding to the subset $\{3\}$;
- 5, corresponding to the subset $\{2, 3\}$.

You want to reconstruct the weights of all the individual items so you can start using your Knapsack algorithm. It might have happened that your friend made a mistake in adding all these weights, so it might happen that her list is not consistent.

Input

- One line containing a single integer $1 \leq n \leq 18$ the number of items.
- 2^n lines each containing a single integer $0 \leq w \leq 2^{28}$, the combined weight of a subset of the items. Every subset occurs exactly once.

Output

Output non-negative integers a_1, \dots, a_n on n lines in non-decreasing order such that $S((a_1, \dots, a_n)) = \{b_1, \dots, b_{2^n}\}$, provided that such integers exist. Otherwise, output a single line containing impossible.

Sample Input 1

1 0 5	Sample Output 1 5
-------------	-----------------------------

Sample Input 2

3 7 5 2 4 1 6 3 0	Sample Output 2 1 2 4
---	---------------------------------------

Sample Input 3

2 0 1 2 4	Sample Output 3 impossible
-----------------------	--------------------------------------

Sample Input 4

2 0 1 1 2	Sample Output 4 1 1
-----------------------	----------------------------------