Thabit Contest #4 - Mirror of PSUT Coding Marathon IX / Binary Circle

Time Limit: 1000ms
Memory Limit: 256MB

I - Binary Circle

Submissions Submit

There are n numbers in a circle **where n is odd**, each number is either 0 or 1, in one operation you can select an index i, and the value at that index and at the two adjacent indices facing opposite of index i on that circle are flipped(if the value was 1 it becomes 0, and if it was 0 it becomes 1). More formally selecting index i will flip the value at indices $i, (i + \lfloor \frac{n}{2} \rfloor - 1 \mod n) + 1, (i - \lfloor \frac{n}{2} \rfloor - 1 \mod n) + 1$.

For example if the numbers are 0010110 and you do an operation on the 6th index, values at indices $6, 2, 3$ will be flipped, and the numbers will become 0100100.

Given the n numbers, print a list of operations that will make all the numbers in the circle equal to 1 using at most $7n$ operations, or state that it's impossible.

Input

The first line contains $n(3 \leq n \leq 9999)$, number of numbers in the circle, n **is odd**.

The second line contains the n numbers, each number is either 0 or 1.

Output

If there is no possible solution using at most $7n$ operations then print "-1" in a single line. Otherwise, let the number of operations be x, print x in the first line and x integers $o_1, o_2, \ldots, o_x$ on the second line, where o_i is the index in which operation i was done on.

Notes

This is how the numbers change in the first example: 01110 $\rightarrow$ 10100 $\rightarrow$ 11111

Samples

Input	Output
5 01110	2 4 2
3 100	-1