Problem L. Binary vs Ternary

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

A *binary string* is a sequence of bits, i.e., "0"s and "1"s. For a binary string S, you can perform the following operations for any number of times:

- Choose a non-empty substring $S[l, r] = S_l S_{l+1} \ldots S_r$, consider it as a ternary (i.e., base 3) integer, and then convert it to the corresponding binary integer. For example, $(101)_3 = (1010)_2$, so you can transform 1**101**10 into 1**1010**10.

Note that the selected substring **may have leading zeros**, but the transformed substring will have no leading zeros. We consider 0 as a proper binary integer without leading zeros. For example, you can transform 01 to 1 because $(01)_3 = (1)_2$. You can also transform 0 to 0 because $(0)_3 = (0)_2$.

Given two binary strings A and B, both starting with the digit "1", you need to determine whether A can be transformed into B in no more than 512 operations. And you need to keep the length of the string no more than 128 during the transformation. If it is possible, print a solution.

Input

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 1000$), indicating the number of test cases.

For each test case, the first line contains the string A ($1 \le |A| \le 64$). The second line contains the string B ($1 \le |B| \le 64$).

It is guaranteed that A and B start with the digit "1" and consist of only "1" and "0".

Output

For each test, if it is impossible to transform string A to B, output "-1".

Otherwise, output an integer n ($0 \le n \le 512$) first, indicating the number of steps. In the next n lines, output two integers l, r ($1 \le l \le r$) indicating the substring you choose in each step. The indices are 1-based. r should be no more than the length of the current string.

Example

standard input	standard output
3 1 111 110110 1101010 1111 111111	-1 1 2 4 2 1 3 2 5

Note

In the first test case, it can be shown that there is no possible solution.

In the second test case, for $A = 110110$, we can choose $l = 2$ and $r = 4$ first. Since $A[2, 4] = 101$, and $(101)_3 = (1010)_2$, so 1**101**10 will be changed to 1**1010**10.