Problem D. Block Edit Distance

Input file:	`block.in`
Output file:	`block.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Recently TopCoder has run a Marathon Match 18 that was intended to support Wikipedia project. The problem that was suggested to the participants had a subproblem — to find a specially defined edit distance between the two words. In this problem we will consider a simplified version of the MM18 problem, but unlike in the Marathon Match we will require the exact solution.

You are given two words S and T. First you are allowed to choose some non-intersecting subwords of S and remove them. Each subword you remove costs you b. Let the resulting word be Z.

After that you must find a standard edit distance between Z and T. To do it, you must find the instruction sequence that converts Z to T. The allowed instructions are "`I`" — insert, "`D`" — delete, and "`C`" — copy.

Consider that you have two pointers, initially the first pointer is at the first character of Z and the second pointer is at the first character of T. "`I`" instruction moves the second pointer one character to the right. "`D`" instruction moves the first pointer one character to the right. "`C`" instruction is applicable only when the two pointers are at equal characters, it moves both pointers one character to the right. Each "`I`" instruction costs i, each "`D`" instruction costs d, each "`C`" instruction costs c.

Find the way to transform S to T in the described way with the smallest cost.

Input

The first line of the input file contains four integer numbers: b, i, d and c ($0 \le b, i, d, c \le 10\,000$). The second line contains S. The third line contains T. The length of each of S and T doesn't exceed 3 000.

Output

The first line of the output file must contain K — the cost of converting S to T in the described way. The second line must contain n — the number of subwords of S to be removed to form Z. The following n lines must contain two integer numbers each — the inclusive ranges of subwords in S to be removed. Characters are numbered from 1.

The last line must contain the sequence of instructions to convert Z to T.

Example

block.in	block.out
3 1 1 0 ABCDEFGHIJKLMN BCDEFZZZZKLM	9 1 7 10 DCCCCCIIIICCCD