

Problem D. Container

Input file: *standard input*
Output file: *standard output*
Time limit: 2.5 seconds
Memory limit: 1024 mebibytes

Cki86201 Container Creator (CCC) makes two kinds of containers. One kind of container has 1-ton capacity, and another one has 2-ton capacity. CCC currently has N containers in a warehouse, which is arranged in a line. To ship the containers, it's good to arrange the containers that are shipped first at the front. Therefore, CCC is going to rearrange those N containers.

To rearrange the containers we use machines. The machine can pick two or three consecutive containers, and reverse their orders. The cost of using the machine is the total capacity of the selected container plus the base cost C . Therefore, reversing two containers $[2, 1]$ takes cost $3 + C$, and reversing three containers $[1, 1, 2]$ takes cost $4 + C$. The number C is given as an input.

Although this task is trivial to cki86201, he has to play Overwatch now. Please find the way of rearranging the containers using the minimum cost.

Input

In the first line, two integers N, C are given. ($1 \leq N \leq 500, 0 \leq C \leq 1000$)

In the next line, a string of size N consisting of two characters '1', '2' is given. This indicates the capacity of the containers stored in the warehouse, in the order of **current** arrangement.

In the next line, a string of size N consisting of two characters '1', '2' is given. This indicates the capacity of the containers stored in the warehouse, in the order of **desired** arrangement.

It is guaranteed that both strings contain an equal number of '1', and equal number of '2'.

Output

In the first line, print the number K denoting the number of times you used the machine.

In the next K lines, print two integers i, j ($1 \leq i < j \leq N, j - i \leq 2$), which indicates that you have reversed the $i, i + 1, \dots, j$ -th containers from the current state, using the machine.

Your answer will be considered correct if the solution is valid (i.e simulating it with the outputted order produces the desired arrangement) and it uses the minimum possible cost.

Examples

standard input	standard output
5 2 11221 21112	2 1 3 4 5
7 0 2212121 1212122	4 6 7 4 6 2 4 1 2
7 2 2212121 1212122	3 1 3 3 5 5 7