Text Search

In this problem, we assign a positive cost to each of the three standard string edit operations Insert, Delete, Rewrite. The costs of different operations may be different. Next, we define the distance between words. The distance of a word B from a word A is equal to the minimum cost of a sequence of operations which transforms A to B. The cost of a sequence of n operations is equal to the sum of costs of all n operations in that sequence. Note that the order of the words is important, the distance of A from B might differ from the distance of B from A.

The task

You are given a text T and a patern P. Find a substring S in T whose distance from P is minimum possible. In case there are more such substrings consider only the shortest ones. If T contains more than one shortest substring with minimum possible distance from P then consider only the leftmost one, i.e. that one which is the closest one to the beginning of T.

Input

The input consists of three text lines. The first line contains five positive integers N, M, cI, cD, cR separated by spaces. The meaning of the integers (in the given order) is the length of the text T, the length of the pattern P, the costs of edit operations Insert, Delete, Rewrite. The second line contains the text T and the third line contains the pattern P. Both T and P use alphabet of small English letters ('a', 'b', ..., 'z'). There are no spaces in the second and in the third input line.

The value of N does not exceed 10^6, the value of M does not exceed 1400, the value of N×M does not exceed 7.2×10^7. The cost of each edit operation is at most 10.

Output

The output contains one text line with three integers separated by spaces. The first integer is the index (position) of the first character of S in T. We suppose that the indices of characters in the text are 0, 1, 2, ..., N−1. The second integer is equal to the length of S. The third integer is the distance of S from P.

It is guaranteed that in the data presented in this problem the substring S always exists and that it is not an empty string.

Example 1

Input

```
6 5 2 1 1
dadbcc
cadba
```

Output

```
1 3 2
```

The substring S in T is 'adb'. The distance of S from P is 2 (apply Delete twice to pattern 'cadba'). There are also other substrings in T whose distance from P is equal to 2, namely 'dadb', 'adbc' and 'dadbc'. They do not represent a solution to the problem as their length is greater than the length of 'adb'.

Example 2

Input

```
10 4 1 3 2
zzzzzyyyyy
yyzz
```

Output

```
0 4 4
```

Example 3

Input

```
18 7 2 9 8
aaaaaaaaabbabbabba
abababa
```

Output

```
8 10 6
```

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

[Link to public data set](#)