

Genetic engineering

A group of scientists tests a new method for creating DNA sequences. The method allows to take basic units, which are a collection of simple DNAs, to modify each of the units by deleting selected nucleotides, and to concatenate the resulting sequences into one DNA sequence. For example, let the basic units be of three types, ACC, AGCTA and CCGGTT. Then, DNA sequence ACACAGCAGCCGT can be created from two basic units of the first type, two basic units of the second type and one basic unit of the third type as

ACC ACC AGCTA AGCTA CCGGTT

where the nucleotides to be deleted are printed in red. Note that each basic unit has its left-to-right order of nucleotides which must be preserved in the created DNA. This means that the leftmost nucleotide of a basic unit can be connected only to the rightmost nucleotide of another unit. It also does not suffice to produce a DNA sequence which (when scanned in the left-to-right order) is the mirror of the sequence to be created.

Each basic unit has to be prepared in advance and has a manufacturing cost represented by a positive integer. In addition, each deletion of a nucleotide increases the cost by 1. The manufacturing cost of using several modified basic units is determined as the sum of their costs (note that concatenations of the units do not increase the cost). For example, let the manufacturing cost of ACC, AGCTA and CCGGTT be 2, 3 and 4, respectively. Then the cost of creating ACACAGCAGCCGT as described above is $(2+1)+(2+1)+(3+1)+(3+4)+(4+2) = 23$.

Nucleotides deletions have an impact on the stability of basic units. For this reason, there is a limit on the number of deletions that can be performed in a basic unit (this limit is the same for basic units of all types). In the example above, we assume that the maximum number of deletions allowed per a basic unit is at least 4.

The task

Let $\Sigma = \{A, G, C, T\}$ be the alphabet of nucleotides. Given a DNA sequence represented by a non-empty string T over Σ , types of available basic units represented by a set of non-empty strings $\{S_1, \dots, S_k\}$, where each $S_i \in \Sigma^+$ is of a manufacturing cost c_i , and a limit D_{\max} on the number of deletions that can be performed over any of the basic units, find a manufacturing plan how to produce T so that the manufacturing cost is the minimum possible. Since the scientists also intend to minimize the number of used basic units, among all the cost-minimum manufacturing plans, find a plan which requires the minimum possible number of used basic units. Note that the number of used basic units is the number of concatenated DNA pieces, not the number of types used in the production. Also note that any amount of basic units of any type is available.

Input

The first input line contains a non-empty string T over Σ , which represents a DNA sequence to be created. The second line contains integers N and D_{\max} separated by space. N is the number of basic unit types, D_{\max} is the maximum number of nucleotide deletions that can be performed per basic unit. Next, there are $2N$ input lines. Each odd line contains a positive integer. This is the manufacturing cost of each basic unit of type represented by a non-empty string over Σ located on the line following after the line with the cost.

It holds $1 \leq |T| \leq 20000$, $1 \leq N \leq 80$ and $1 \leq D_{\max} \leq 40$. Each basic unit type is of length at most 100 and of cost at most 500.

Output

The output consists of one line which contains two integers, P and U , separated by space. The first integer is the minimum manufacturing cost of T , the second integer is the minimum number of basic units needed to produce T with manufacturing cost P . It is a guaranteed that a feasible solution always exists.

Example 1

Input

```
ACACAGCAGCCGT
3 4
2
ACC
3
AGCTA
4
CCGGTT
```

Output

```
23 5
```

Example 2

Input

```
CACCATCACCATACCCTT
4 1
10
CACCTT
10
CGTAGTG
14
CACCCAT
20
```

CACACAT

Output

41 3

An optimal solution is: CAC^{CC}CAT CAC^{CC}CAT C^{ACC}CCTT

Example 3

Input

CCGTCTTCCCTTGAAC^TCCGTCCCCCG

6 2

2

TTC

5

CCCG

6

GAAT

14

GAAC

4

CCT

3

CGT

Output

46 11

An optimal solution is: CC^{CG} ^{TTC} TTC CCT ^{TTC} GAAT ^{TTC} ^{TTC} CGT CCT ^{CCCG}

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](#)