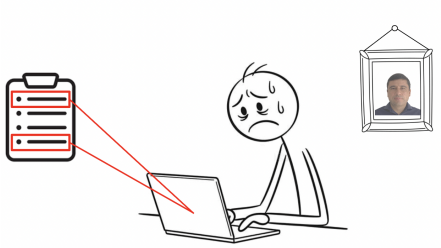
Procrastination Optimization



You want to write a program that will allow you to optimize the Kattis problems you select when completing your COMP 321 assignments. For each assignment, you only need to answer a subset of the listed questions to reach full marks. Each problem has a:

- Point value (some positive integer)
- Difficulty rating (an integer from 5.0 to 10.0)
- Topic (like trees, graphs,...)
- Text length (number of words)

Although you're a lazy student, you still care about your grades, and so you want to achieve full marks on your assignment while putting in the least amount of effort possible. On a given assignment, you need to attain at least M points, and you want to do so while trying to:

1. Minimize the total difficulty of the chosen problems first, then
2. Minimize the number of problems solved. Then,
3. Align the topics of the problems with concepts you have the best understanding of. Then finally,
4. Minimize the total length of text read

These requirements are listed in preferential order. That is, you first and foremost want to minimize the difficulty of questions you select. If there are still some choices for question subsets, then you can narrow it down further by minimizing the number of points solved, and work down to the least pressing requirement, which is minimizing the total length of text you read.

Input

The first line contains two integers M and N, representing the minimum number of points required for full marks and the number of available problems on this assignment, where $1 \leq N \leq 60$, and $1 \leq M \leq 10^{15}$.

The second line contains a space-separated list of five distinct strings, each representing a topic. The list is ordered from the topic you understand the best to the one you understand the least.

The following N lines describe the problems. Each problem is given in the format:

`Problem_No.` `Points` `Difficulty` `Topic` `Text_Length`

Here:

- `Problem_No.` is an integer in $[1, N]$.

- `Points` is a non-negative integer representing how many points the problem is worth. (It is not required to be $\leq M$.)
- `Difficulty` is an integer rating from 5 to 10, with larger values indicating harder problems.
- `Topic` is a non-empty string (no spaces) indicating the assigned topic of the problem. Every topic is chosen from one of the five topics in the second line of input.
- `Text_Length` is an integer between 1 and 1000 representing the number of words in the problem statement.

Assume that each problem will be guaranteed to have at least 10% of the minimum point value M needed to achieve full marks.

Output

One line, containing the problem numbers, separated by spaces, that you choose in order to complete your assignment under the above conditions. Assume there is *always* a subset of questions in the problem list where it is possible to achieve full marks.

Sample Input 1

```
10 4
dp graphs arrays trees queues
1 5 8 dp 120
2 6 10 graphs 200
3 4 6 arrays 50
4 8 9 dp 300
```

Sample Output 1

```
3 4
```

Sample Input 2

```
10 3
stacks queues trees strings dp
1 2 5 stacks 100
2 2 5 stacks 100
3 10 10 trees 100
```

Sample Output 2

```
3
```

Sample Input 3

```
1000000000000000 4
greedy dijkstra strings stacks dp
1 600000000000000 8 greedy 5000
2 500000000000000 7 dijkstra 8000
3 400000000000000 6 strings 2000
4 700000000000000 10 greedy 10000
```

Sample Output 3

```
1 3
```