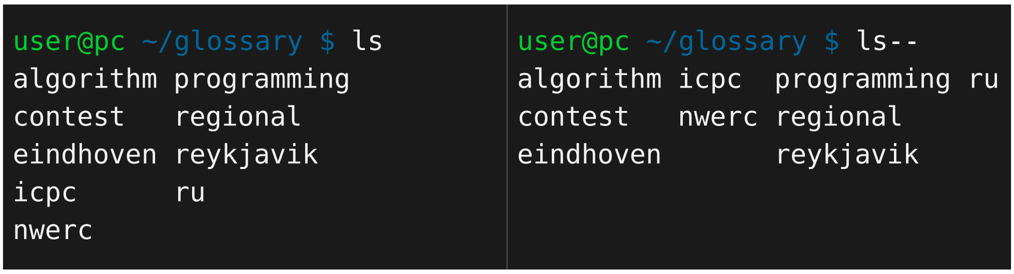
Problem G
Glossary Arrangement
Time limit: 5 seconds

In Unix based operating systems, one of the most frequently used commands is `ls`, which displays a list of all the files in a directory in lexicographic order. In the most basic form, `ls` prints each filename on a separate line, but to improve readability and save screen space, the list is usually split into multiple columns which are displayed side by side.

A friend of yours is writing a book about NWERC and has put you in charge of editing the glossary of relevant terms at the end of each chapter. Each glossary must use the same multi-column layout as `ls`, so you decided to go for the lazy option: For each word in the glossary, you created an empty file with that name, and simply let `ls` do the heavy lifting.

Unfortunately, your friend is not satisfied with your layouts and complains that some of them take up too much vertical space. The problem with your approach is that `ls` always forms columns of equal height, except for the last column, which may be shorter. This sometimes ends up using more lines than would be needed if the column heights could be chosen more freely:



```
user@pc ~/glossary $ ls
algorithm programming
contest   regional
eindhoven reykjavik
icpc      ru
nwerc
```

```
user@pc ~/glossary $ ls--
algorithm icpc  programming ru
contest   nwerc regional
eindhoven       reykjavik
```

Figure G.1: Saving two lines using variable column heights.

Begrudgingly, you decide to write your own improved version of `ls`, `ls--`, that also displays the contents of a directory in lexicographic order, but uses variable column heights to always achieve the lowest possible number of lines.

Columns have a fixed width, which is the length of the longest filename within the column, and are separated by a single space. The names in each column must be left-aligned and padded to the column width using spaces. Also, the terminal you are using has a fixed width of w characters which the table may not exceed.

Given the contents of a directory as a list of filenames, find an optimal column layout that minimizes the number of lines needed to print the entire list.

Input

The input consists of:

- One line with two integers n and w ($1 \leq n, w \leq 5\,000$), the number of files and the width of the terminal.
- n lines, each with one filename s ($1 \leq |s| \leq w$, s consists of lowercase English letters).

The filenames are distinct and in lexicographic order. The total number of letters is at most 10^6.

Output

Output an optimal way of listing the given filenames:

- One line with two integers r and c, the number of lines and the number of columns used.
- One line with c positive integers $a_1, a_2, \ldots, a_c$, the widths of the columns.
- The table of filenames, subject to the following formatting:
  - There are c columns, where column i has width a_i for each i, and within each column there are at most r filenames that are aligned on the left and grouped at the top.
  - The filenames are aligned using spaces, with exactly one space between columns.
  - The total width of the table is at most w.
  - When reading column by column, the filenames appear in lexicographic order.

Note that unlike in other problems, you strictly need to follow the above formatting rules for whitespace. However, we still allow trailing whitespace at the end of each line, even if this whitespace exceeds the width w.

Sample Input 1

```
9 30
algorithm
contest
eindhoven
icpc
nwerc
programming
regional
reykjavik
ru
```

Sample Output 1

```
3 4
9 5 11 2
algorithm icpc  programming ru
contest   nwerc regional
eindhoven       reykjavik
```

Sample Input 2

```
6 10
aaa
bb
ccccc
ddd
eeeee
fffff
```

Sample Output 2

```
4 2
3 5
aaa ccccc
bb  ddd
    eeeee
    fffff
```

Sample Input 3

```
5 15
pppp
ppppp
pq
pqab
xyzff
```

Sample Output 3

```
2 3
5 2 5
pppp  pq pqab
ppppp    xyzff
```