F. Matching Names

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Teachers of one programming summer school decided to make a surprise for the students by giving them names in the style of the "Hobbit" movie. Each student must get a pseudonym maximally similar to his own name. The pseudonym must be a name of some character of the popular saga and now the teachers are busy matching pseudonyms to student names.

There are n students in a summer school. Teachers chose exactly n pseudonyms for them. Each student must get exactly one pseudonym corresponding to him. Let us determine the relevance of a pseudonym b to a student with name a as the length of the largest common prefix a and b. We will represent such value as . Then we can determine the *quality* of matching of the pseudonyms to students as a sum of relevances of all pseudonyms to the corresponding students.

Find the matching between students and pseudonyms with the maximum *quality*.

Input

The first line contains number n $(1 \le n \le 100\,000)$ — the number of students in the summer school.

Next n lines contain the name of the students. Each name is a non-empty word consisting of lowercase English letters. Some names can be repeating.

The last n lines contain the given pseudonyms. Each pseudonym is a non-empty word consisting of small English letters. Some pseudonyms can be repeating.

The total length of all the names and pseudonyms doesn't exceed $800\,000$ characters.

Output

In the first line print the maximum possible *quality* of matching pseudonyms to students.

In the next n lines describe the optimal matching. Each line must have the form a b $(1 \le a, b \le n)$, that means that the student who was number a in the input, must match to the pseudonym number b in the input.

The matching should be a one-to-one correspondence, that is, each student and each pseudonym should occur exactly once in your output. If there are several optimal answers, output any.

Examples

input

```
5
gennady
galya
boris
bill
toshik
bilbo
torin
gendalf
smaug
galadriel
```

output

```
11
4 1
2 5
1 3
```

```
5 2
3 4
```

Note

The first test from the statement the match looks as follows:

- **bil**l → **bil**bo (lcp = 3)
- **gal**ya → **gal**adriel (lcp = 3)
- **gen**nady → **gen**dalf (lcp = 3)
- **to**shik → **to**rin (lcp = 2)
- boris → smaug (lcp = 0)