COP3530

Project #2

The Hidden Room of AGCA

AGCA

You are part of a group called AGCA[1]. In your first meeting it was decided that the group members should never meet in person again. However, it was also decided that a Hidden Room should be built and secured to make the members able to share what they want to share with other group members. You were the one selected to make sure the room is secure from any intrusion, while being accessible to group members.

Door Lock

You designed a door lock for the hidden room that would open the door if the correct combination is entered. However, in order to make sure nothing is gained by anyone trying to infiltrate in the room, the door lock is set up in a way that would blow up the whole room if the entered combination is wrong.

In order to open the door one has to enter N numbers into the door lock, and as long as the numbers include the correct combination in the correct order, the entered numbers are accepted and the door will be opened. For example, if the combination is **6 18 9**, then entering 0 **6** 10 9 **18** 3 **9** 5 will open the door, because it includes the combination in the correct order.

Combination

It is generally known that a password should be changed regularly. However, since the members of the group are never going to meet again, there is no way to give them the new password, i.e. new combination, if it is ever changed.

In order to overcome this problem, you decided to give half of the information about the combination to the members and make the other half part of the public information. By doing this, the combination can be changed at any given time, since you have control over the public part, on the other hand no one other than the members can find the combination, since they don't have access to the private information shared with the members.

You implemented this idea by giving an N by N matrix of alphabetical characters to each member as the private information, and then by showing a hint string on the door lock they can find out about the correct combination as follows: the ith number that members will enter is the number of occurrences of the hint string in the ith string in the matrix, including overlapping occurrences. For example if the 2nd string in the matrix is “bababc” and hint string is “bab” the 2nd number that members will enter will be 2, or if the 4th string in the matrix is “babbbbbcb” and the hint string is “bbb” then the 4th number that members will enter will be 3. Not that as it is shown by the examples above the hint should be found in consecutive elements of the string, for example there is no occurrence of “ab” in “azb”.

[1] AGCA stands for “A Group Called AGCA”.

Mistake

After the meeting is concluded you decide to find a hint and set up the combination for the door lock. But you suddenly realize that you have made a grave mistake. You have forgotten to tell the members whether the i^{th} string in the matrix refers to the i^{th} row or the i^{th} column. At first you decide to set the combination to 0 and set up the hint string in a way that does not occur anywhere in the matrix. However this will make it very easy for non-members to enter the room, since the door will be opened as long as the numbers they enter contains a 0. Then you decide to assume that everyone will consider i^{th} string to be the i^{th} row, however you remember that the door lock is programmed to blow up in case of the wrong combination and this will put the lives of group members in danger.

Solution

In order to find a way to solve this problem, you have to write a program that would get the matrix and the hint string and will print the maximum length combination that you can safely set up on the door lock with that hint string, i.e. a combination that has the maximum length among the combinations that would not put the lives of the members in danger, regardless of their assumption about the meaning of i^{th} string.

Problem Definition

Input

Your program should read the hint string and the matrix from the input.

The first line of the input will contain the hint string. The second line of the input will contain *N*. The following *N* lines, each, will contain a string of length *N* denoting the rows of the matrix. For example for the hint string “*abc*” and the following matrix

$$\begin{bmatrix} a & c & b & a & b \\ b & b & a & b & c \\ a & b & c & c & a \\ b & a & c & a & b \\ a & b & b & a & b \end{bmatrix}$$

the input will be formatted as follows:

```
abc
5
acbab
bbabc
abcca
bacab
abbab
```

Output

Your program should output a combination with maximum length, based on what was described in the previous sections. The combination should be printed in one line with its numbers separated by spaces. For example, a possible solution for the above input is:

```
0 1 0
```

Notes

1. As a string, the rows of the matrix are read from left to right, and the columns of the matrix are read from top to bottom.
2. Your solution should perform in $O(N^2)$.
3. The length of hint string is not constant, i.e. it is not $O(1)$, but it is at most than *N*.
4. You can assume that the maximum length is at least 1, i.e. you can always find a combination of length 1 that will work.

What to Submit

You should submit your source code for solving the problem, along with a PDF file. In the PDF document you should briefly, i.e. 1 paragraph, explain your solution and also prove that your solution performs in $O(N^2)$. In your proof you can assume the run-time of well-known algorithms, i.e. you do not need to copy any proofs from Wikipedia.