COP3530

Project #2

The Hidden Room of TTS

TTS

You are part of the TTS society[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 secured against 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 person who entered the wrong combination.

The door lock's combination is a sequence of numbers and in order to open the door one has to enter a sequence of numbers into the door lock that has the combination in the correct order, i.e. the entered sequence can have extra numbers that are not part of the combination too. 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, but entering 0 9 18 will blow up the person since it does not contain 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 needs to be a mechanism that would let the other members know what to enter after the combination is changed.

In order to overcome this problem, you decided to give some initial information to the members and add a display to the lock that would show some other information in a way that would allow the members find a good sequence. By doing this, after changing the combination you can always change the part that is shown on the lock and the members will be able to find a new sequence based on the initial information that they have. However, someone who is not part of the group will not be able to open the door since they do not have access to the initial information.

You implemented this idea by giving a string of *N* alphabetical characters to each member as the initial information and showing another alphabetical string as a hint on the door lock. The members can find a good sequence in the following manner: the i^{th} number of the sequence that the members should enter is the difference between the index of i^{th} and $(i+1)^{th}$ occurrence of the hint string in the given initial string. For example if the initial string is "babaabac" and the hint string is "ba" the indices of the occurrences are 0 ("**ba**baabac"), 2 ("ba**ba**abac"), and 5 ("babaa**ba**c"). So the differences in the indices would be 2 – 0 = 2 and 5 – 2 = 3. Or if the initial string is "aaabaaa" and the hint string is "aa" then the indices are 0, 1, 4, and 5, so the differences are 1, 3, and 1. Note that as it is shown by the examples above, the

1 TTS stands for "The TTS Society".

hint should be found in consecutive elements of the string, for example there is no occurrence of “ab” in “azb”.

Misunderstanding

After the meeting is concluded you find out that some of the members have misunderstood the method of finding the sequence and will enter the numbers from right to left, instead of from left to right. For example, if the indices were 0, 1, 3, 6, and 10, then some of the members will enter the sequence 1 2 3 4 and some of them will enter the sequence 4 3 2 1. This means that if the combination is not included in one of the them, it would put the life of some of the members in danger.

Solution

Considering how the lock combination works, i.e. the entered sequence may contain extra numbers that are not part of the combination, there is the possibility to choose the combination in a way that would allow all of the group members open the door. For example, if the indices are 0, 2, 3, 6, and 8, then some of the members will enter 2 1 3 2 and some of the will enter 2 3 1 2. This means that if the combination is **2 1** then whether you enter **2 1** 3 2 or **2** 3 **1** 2, the lock will open since the combination is included in the entered sequence in the correct order in both possibilities.

However, in order to increase the security as much as possible, you want to find the longest such combination. In other words, the longest combination that would allow the members enter the room whether they enter the numbers from left to right or right to left. For instance, in the previous example that we had, 2 1 3 2 and 2 3 1 2, you can set the combination to **2 1 2** which is longer that **2 1** and would still allow all of the members open the door: **2 1** 3 **2** and **2** 3 **1 2** both contain **2 1 2** in the correct order.

Problem Definition

Input

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

The first line of the input will contain the initial string. The second line of the input will contain the hint string.

So, if the initial string is “ababbdabcabab” and the hint string is “ab” the input will be formatted as follows:

ababbdabcabab
ab

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:

2 4 2

Notes

1. As it is seen in the example there may be multiple correct outputs. For the given example 2 3 2 is also an acceptable answer, but 2 1 is NOT an acceptable answer since it’s not the longest.
2. If the length of the initial string is N, the length of the hint string is W, and the number of occurrences of hint in the initial string is M, your solution should work in $O(N+W+M^2)$.
    - Warning: if **str** the initial string and **hint** is the hint string, then **str.find(hint)** will run in $O(NW)$, i.e. it’s NOT $O(N+W)$, and consequently it’s not $O(N+W+M^2)$.
3. If there are no index differences, i.e. no occurrences or just one occurrence, you should print and empty line.

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 show that your solution performs in $O(N+W+M^2)$. 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.