Editorial- Sam's Encryption

Here, they give the string & list of **n** keywords of equal length. you have to find the a list of indexes of the starting position of possible secrets. The output should print each index of the index list in separate lines. Indexes should print in ascending order. If there are no possible indexes then print -1.

First, you have to make the frequency map for the keywords. Frequency map means no of occurences of a string/int. Here the password length is equal to the sum of the given keywords length.

Then, you have to divide the given string into substring with the password size.

Then for each substring ***P*** in the ***S*** with the length of the password length,

1. Divide ***P*** into substrings(starting from index ***0***) with the size of the given keywords size.
2. Make another frequency map from these substrings.
3. If the new frequency map and the frequency map of the given keywords are identical output the starting index of the ***P***.

For example: "**satpipcabpipsat**" is the string that given you. sat-1, pip-1 ,cab-1 occurrences are there in the given keywords. Here, in the substring starting from index 0 (**satpipcat**) and in the substring starting from index 6 (**cabpipsat**) you can find the same frequency for given keywords, so those two are possible secrets.

Time complexity will be the **O(N*M*W)** Where ***N*** = length of the string ***S***, ***M*** = number of keywords, ***W*** = length of a keyword. Here loop should run till ***length of string - (length of word * no. of words)*** as after that no such combination will be possible due to less no. of characters respective to that needed to make a secret from given keywords.