Ancient cryptography (crypto)

In the town of Visoko, archaeologists are trying to decipher an ancient inscription found in the Bosnian pyramids. They have found a very long inscription written on a stone, consisting of a very limited set of unknown symbols. To make things even worse, all of the previously found inscriptions were encrypted with ancient “Pharaoh+” encryption.

Luckily, the team of cryptologists discovered that the “Pharaoh+” encryption is easily deciphered as long as one has an adequate *key*. Even better, the key is a word from ancient language spoken in Visoko, and it is contained in the encrypted inscription at unknown locations.

Thus, cryptologists and archaeologists would like to ask you to help them find out the key in the inscription. The key is the longest word from the dictionary found in the inscription (it seems that a longer key always meant better encryption). If there are multiple longest words in the inscription, the one that occurs first in the inscription should be chosen as a key.

Task

Your task is to write the procedure GetKey that takes seven parameters. The first one is an integer n – the number of letters in the alphabet. The second parameter is a string representing the alphabet. The letters in this alphabet are unique. For your convenience, archaeologists replaced each symbol with an uppercase Latin letter. The third parameter is an integer *m* – number of words in the dictionary. The fourth parameter is array of strings representing words from the dictionary of ancient language (each array member represents one word from the dictionary, with maximum length *w*). The fifth parameter is a string representing the inscription (with length x). Sixth and seventh parameters are *idx* and *count* that are passed by reference (Pascal) or as pointers (C/C++). You should write index of the word in the dictionary (first word in the dictionary has index 0) that was matched as the key to *idx* and number of key’s occurrences to *count*. If a key cannot be found, you should write -1 to *idx* and the value written to *count* is ignored.

Words in the dictionary and the inscription will not contain any other characters than those given in the alphabet.

Example

If we have an alphabet of five letters “ABCDE”, and five words in the dictionary: “AA”, “BBBB”, “CDCD”, “AAB” and “EEEEE”, and given the inscription “ACCCAAAABBBBCDCDCDBBBB”, result is 1 (index of word “BBBB”) and *count* is 2.

Word “BBBB” is the longest one found in the inscription, and there are 2 occurrences of “BBBB”. Although “CDCD” is also found 2 times in the inscription, the first word of length 4 occuring in the inscription is “BBBB”. “EEEEE” does not appear in the inscription, even though it is the longest word in the dictionary.

Subtasks

Subtask 1 (6 points) : n<=26, m<=5, w<=40, x<=35,000

Subtask 2 (22 points) : n<=26, m<=65, w<=100, x<=3,500,000

Subtask 3 (23 points) : n<=10, m<=65, w<=15, x<=60,000,000

Subtask 4 (24 points) : n<=10, m<=25,000, w<=50, x<=100,000,000

Subtask 5 (25 points) : n<=5, m<=5,000,000, w<=15, x<=100,000,000, additionally the amount of characters in the dictionary won’t exceed 70,000,000.