Programming Assignment #3

Suppose you were working on a language learning module of an artificial intelligence project. The first goal of your module was to learn the language words by going through a large set of transcripts. However, for some unknown reason, while generating the text, the transcribing program omitted all the white spaces. Thankfully though, there exists a dictionary that contains all the allowed words of this language. Thus, you must write a pre-processor to parse the transcript text, and use the dictionary to identify the legal words of the language by inserting white-spaces in the text. Let us look at a small example. Suppose the language being learnt is English, and the transcript text — without white-spaces — is: insearchoflosttime. Then, one possible set containing five acceptable words will be: {in, search, of, lost, time}.

In this assignment, you need to implement one such pre-processor. The language to be learnt however is not necessarily English, but some common words may exist.

Part 1: Programming Task [80 points]
Your program will read input from an input file and write to standard output. It should read the transcript text, supplied by the input, into one string called `str`. It will store a set of allowed language words supplied by a dictionary, which is also supplied by the input, and add white-spaces to *str* to construct sentences such that all the words in any sentence are allowed in the language.

Input Format:
The first line of the input file contains a non negative integer n, followed by n lines containing one legal language word per line. The last line in the file contains the transcript string, `str`, without any white space. Note that all the n words need not be a substring of `str`.

Output Format
The first line of your output should print k — a non-negative integer that is the number of sentences produced by your program. If k is zero, then your program should not output anything else. If k is non-zero, then your program should output k more lines, containing one valid sentences per line. Refer to the samples at the end of this document.

Part 2: Report [20 points]
Write a short report on following parts:

(a) Describe your approach and write a report of all the algorithmic information including but not limited to the time complexity, approach and the pseudocode.

(b) Prove that your algorithm is guaranteed to provide the correct result given a valid input. Explain how you validate and verify your algorithm.

(c) **Extra Credit (5 points)** Is there a loop invariant in your algorithm? If there is one, explain what the loop invariant in your algorithm is and why.

Important: Adhere to the general submission guidelines for programming assignments regarding the files to be submitted, name of executable, and instructions that need to be supplied for compiling/execution.

1 Samples

1.1

Input:
4
algor
algo
ithm
rithm
algorithm
Ouput:
2
algo rithm
algor ithm

1.2

Input:
5
bar
bars
and
sand
doll
barsanddoll
Ouput:
2
bars and doll
bar sand doll

1.3

Input:
6
a
alan
lan
lant
uring
turing
alanturing
Output:
3

a lant uring
a lan turing
alan turing