Instructor Name: Siddhartha Nath
Email: s.nath@northeastern.edu

Northeastern University

Program Structures and Algorithms (INFO 6205)
Programming Assignment 2 – 100 points

Assignment 1 (100 points)**.** *You are given a* ***target*** *string and a* ***wordbank*** *(which is a 1D array of strings). You will write a program based on* **dynamic programming** *algorithm that will return a 2D array containing* ***all of the ways*** *that the "target" can be constructed by concatenating elements of the "wordbank" array. A single element of this output 2D array should represent one of the combinations that constructs the "target". You may reuse elements of the "wordbank" as many times as you need.*

Your program will take as input two arguments. The first argument is called "–target" and will contain "target" string. The second argument is called "–wordbank" contains a list of strings with spaces that you will read to form your 1D "wordbank" array. For every run of your program, you must report the runtime in seconds. You could use package time in Python.

Your output should be as follows.

Number of ways: N
List each of the ways, one in each line as $N \times K$-dimensional array, where N is the number of ways, and K are the elements from the wordbank used.
Runtime in seconds.

Example 1: *If the target string is "purple", and the wordbank list of strings are: ["purp", "p", "ur", "le", "purpl"]. The output will be as follows.*

your_program.py –target purple –wordbank purp p ur le purpl

Number of ways: 2
[
[*"purp", "le"*]
[*"p", "ur", "p", "le"*]
]
Runtime: YYY seconds

Example 2: *If the target string is "hello", and the wordbank list of strings are: ["cat", "dog", "mouse"]. The output will be as follows.*

your_program.py –target hello –wordbank cat dog mouse

Number of ways: 0
[
]

Runtime: YYY seconds

Example 3: *If the target string is "" (empty string), and the wordbank list of strings are: ["cat", "dog", "mouse"]. The output will be as follows.*

your_program.py –target '' –wordbank cat dog mouse

Number of ways: 1
[
[ ]
]
Runtime: YYY seconds

You must submit the following to receive full credit.

(1) (20 points) Clearly state your subproblem definition, decisions and recursion as comment in the beginning.

(2) (40 points) Python or Java script or C++ file with comments and structure.

(3) (15 points) At least one testcase on which you validated your program. Please specify your testcase(s) in one testcase.txt file. Each line in the file **MUST** *include the format as follows:*

your_program.py –target $target_string –wordbank $list_of_strings

(4) (5 points) Please make sure you handle corner cases, and gracefully error out when you are provided with incorrect inputs, etc. Program crashes or errors or no outputs will be penalized.

(5) (20 points) For Example 1 above, please show the execution of your code by printing the solution to each subproblem in the order they're solved.