

**Homework #12**

**01286121 Computer Programming**

**Software Engineering Program,**

**Department of Computer Engineering,**

**School of Engineering, KMITL**

By

66011149 Phatthadon Sornplang

1. **To save time and space when sending an SMS or a tweet, some words or phrases are often abbreviated.**
   1. Write a function textese(s) which, given a string s of message in plain English, returns a string resulted from replacing words or phrases in s using the above abbreviations.
   2. Write a function untextese(s) which, given a string of s message employing the above abbreviations, returns a string of message in plain English.

abb = {"be": "b", "because": "cuz", "see": "c", "the": "da", "okay": "ok", "are": "r", "you": "u",

"without": "w/o", "why": "y", "see you": "cu", "ate": "8", "great": "gr8", "mate": "m8",

"wait": "w8", "later": "l8r", "tomorrow": "2mro", "for": "4", "before": "b4", "once": "1ce",

"and": "&", "Your": "ur", "You're": "ur", "As far as I know": "afaik", "As soon as possible": "ASAP",

"At the moment": "atm", "Be right back": "brb", "By the way": "btw", "For your Information": "FYI",

"In my humble opinion": "imho", "In my opinion": "imo", "Laughing out loud": "lol", "Oh my god": "omg",

"Rolling on the floor laughing": "rofl", "Talk to you later": "ttyl"}

def textese(txt):

new = txt

for k in sorted(abb, key=len, reverse=True):

new = new.replace(k, abb[k])

return new

txt1 = 'For your Information Imma be back later'

txt2 = 'I am Rolling on the floor laughing'

txt3 = "In my humble opinion, You're so cute, I want to Talk to you later"

print(textese(txt3))

def untextese(s):

reversed\_abb = {v: k for k, v in abb.items()}

words = s.split()

new = []

for word in words:

if word in reversed\_abb:

new.append(reversed\_abb[word])

else:

new.append(word)

return ' '.join(new)

txt4 = "FYI Imma b back l8r"

txt5 = "I am rofl"

txt6 = "imho , ur so cute, I want to ttyl"

print(untextese(txt4))

print(untextese(txt5))

print(untextese(txt6))

A computer screen shot of a black background

Description automatically generated

1. Given two dictionaries dict1 and dict2, suppose we define the composition of dict1 and dict2 to be the dictionary dict3 such that s (key:value)-pair k:v is in dict3 if and only if there a exists some object m such that k:m is in dict1 and m:v is in dict2

dict1 = {}

dict2 = {}

def composite(dict1, dict2):

dict3 = {}

for k1, v1 in dict1.items():

for k2,v2 in dict2.items():

if v1 == k2:

dict3[k1] = v2

return dict3

dict1 = {'a':'p', 'b':'r', 'c':'q', 'd':'p', 'e':'s'}

dict2 = {'p':'1','q':'2','r':'3'}

print(composite(dict1, dict2))

A computer screen with text

Description automatically generated

1. Suppose we are given sets s and t. The cartesian product of s and t is the set of all tuple(x,y) such that x is a member if s and y is a member of t.

Write a python function product(s1,…, sN) where s1, …, sN are sets and N>= 1, which returns the cartesian product of s1, … , sN

def product(\*sets):

if not sets:

return set([])

if len(sets) == 1:

return set([(item,) for item in sets[0]])

sub\_product = product(\*sets[1:])

cartesian = [(item,) + tuple\_ for item in sets[0] for tuple\_ in sub\_product]

return set(cartesian)

s1 = set([1,2,3])

s2 = set(['p','q'])

s3 = set(['a','b','c'])

print(product(s1,s2))

print(product(s1, s2, s3))

print(product(s1))

