Computational Thinking and Problem Solving (COMP1002) and Problem Solving Methodology in Information Technology (COMP1001)

Assignment 2

ZHANG WengYu 21098431d

Q1.

a)

Because after the **core encryption process: (ord(p_i) + ord(k_j) – 2 * ord('a'))**, the result value may be greater than 26 which is the maximum index of the English letter('z'). For example, English text[i]: 'n', key[i]: 'p', then the result is 28 which is great than 26, without "mod 26", the next process cannot return an English text according to the ACSII code.

With "mod 26", when the result value from the **core encryption process** is greater than 26('z'), it can back to 0('a') and start again. For the former example, 28 mod 26 results in 2, with the next process (+ ord('a')), 'c' can be returned correctly.

b)

Input: ***p*** and ***k***
Set ***c*** as an empty list
Set ***i*** as 0
Repeat
    Set ***pi*** to ***i*** th letter of ***p*** from left to right.
    Set ***kj*** to ***(i mod (length of k))*** th letter of ***k*** from left to right.
    Set ***ci*** to chr((ord(***pi***) + ord(***kj***) - 2*ord('a')) % 26 + ord('a')),
    and append ***ci*** to ***c***.
    Add 1 to ***i***.
Until ***i*** is greater than the length of ***p***
Convert ***c*** to String
Output: ***c***

ZHANG WengYu 21098431d

Q1.

c)

Input: ***c*** and ***k***
Set ***p*** as an empty list
Set ***i*** as 0
Repeat
    Set ***ci*** to ***i*** th letter of ***c*** from left to right.
    Set ***kj*** to ***(i mod (length of k))*** th letter of ***k*** from left to right.
    Set ***pi*** to chr((ord(***ci***) – ord(***kj***) + 26) % 26 + ord('a')),
    and append ***pi*** to ***p***.
    Add 1 to ***i***.
Until ***i*** is greater than the length of ***c***
Convert ***p*** to String
Output: ***p***

Q2.

a)

Input: the number of tiles ***m***, some tiles have coin in it, and the index ***t*** of target tile that move all coins to
Set list ***l*** to contain indexes of the tile that placed a coin in it, from left to right, starting from 0
Set ***count*** to 0
Repeat get element of the ***l*** from left to right, as ***i***
    Get the absolute value of (***t*** subtracts ***i***)
    Add that value to ***count***
Until all elements are got from ***l***
Output the total moving distance: ***count***

b)

Input: the number of tiles ***m***, some tiles have coin in it
Set list ***l*** to contain indexes of the tile that placed a coin in it, from left to right, starting from 0
Set ***t*** as the index of target tiles that move all coins to
Set ***t*** to the floored quotient of ***l***'s length and 2
Output the moving distance of list ***l*** with target ***t*** using the function in a)

ZHANG WengYu 21098431d