

ASSIGNMENT 2

JIANG, Guanlin (21093962D)

Q1:

- a) Because the `ord()` this function will be let letter into the ASCII code format, also the English alphabet have 26 letters, but because of use the formula ($\text{ord}(p_i) + \text{ord}(k_j) - 2 * \text{ord}('a')$) sometimes the answer will be over 26, so use mod 26 can make sure the value between or equal between 0 and 25 (the value of the lower letter after this formula must between $0 \leq \text{answer} \leq 25$).

Example 1 (not over 26):

P = "a"

K = "c"

$\text{answer} = \text{ord}("a") + \text{ord}("c") - 2 * \text{ord}("a")$

$\text{answer} = 97 + 99 - 2 * 97$

$\text{answer} = 2$

because $0 < \text{answer} < 26$,

so, $2 \bmod 26 + 97$ # remind: $\text{ord}("a") = 97$

C ASCII code = $2 + 97 = 99$

So, C = "c"

Example 2 (over 26):

P = "u"

K = "p"

$\text{answer} = \text{ord}("u") + \text{ord}("p") - 2 * \text{ord}("a")$

$\text{answer} = 117 + 112 - 2 * 97$

$\text{answer} = 35$

because $\text{answer} > 25$,

so, $35 \bmod 26 + 97$ # remind: $\text{ord}("a") = 97$ (so use mod to make sure the value is between that range)

C ASCII code = $9 + 97 = 106$

So, C = "j"

b)

Input: p and k

Output: c

Set p = input English text

COMP1002 ASSIGNMENT 2

Set k = input the Key

p_list = all p letter into the list

k_list = all k letter into the list

for i in range of length of p_list:

 if length of k_list < length of p_list:

 j = I mod length of k_list

 for p_list and k_list in ASCII code:

 p_ascii = p_list[i] in ASCII code

 k_ascii = k_list[j] in ASCII code

 answer = (p_ascii + k_ascii - 2 * 97) # ord('a') mod26 + ord('a') = 97

 set c_answer = 0 # create a variable c_answer

 if answer > 26:

 then answer = answer mod 26

 else:

 answer will be not change

 c_answer = answer + 97

 for c_answer in ASCII code:

 c = the letter to display the c_answer which shows like ASCII code

return c

c)

Input: c and k

Output: p

Set c = input the Encrypted text

Set k = input the Key

c_list = all c letter into the list

k_list = all k letter into the list

for i in range of length of c_list:

 if length of k_list < length of c_list:

COMP1002 ASSIGNMENT 2

```
j = I mod length of k_list
for c_list and k_list in ASCII code:
    c_ascii = c_list[i] in ASCII code
    k_ascii = k_list[j] in ASCII code
set p_answer = 0
p_answer = ((c_ascii - 97) + 97 * 2 - k_ascii) # ord('a') mod26 + ord('a') =
97
if answer < 97:
    then answer = answer + 26
else:
    answer will be not change
for p_answer in ASCII code:
    p = the letter to display the p_answer which shows like ASCII code
return p
```

Q2:

a)

Input: The number of square tiles which are user want to move the coin.

Output: The distance which is total of all the coins move to the square tiles.

set a is the number of square tiles that user want to coin move to those tiles

set a = 0

for I in each square tile:

mark tiles [0...M] and find which tiles have coin

repeat

moving the coin to a

counting the distance of the coin moving

until all the coins are already moved and counted

set total_distance = the distance to be counted

set a = total_distance

COMP1002 ASSIGNMENT 2

return a

b)

input: The coin user wants to move

output: The total distance which is minimum answer

use the solution in 2a) as a function to be function in here

set answer_list = [] # which is empty list

for j in each square tile:

 use the function to return the calculated answer to answer_list

set s_d is maximum vaule

for distance in answer_list:

 if minimum distance < s_d:

 set s_d = distance

return s_d