

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

## Assignment 2 (Due on 11 November 2021 (Thu) at 12:00 noon)

- [25 marks] In cryptography, a polyalphabetic substitution cipher is an encryption algorithm that converts an English letter to another letter based on a key. For example, if the English text ( $p$ ), is “dennisliu” and the key ( $k$ ) is “comp”, the encrypted text ( $c$ ) will be “fszckgxxw”. The following table shows the conversion:

English text ( $p$ )	d	e	n	n	i	s	l	i	u
Key ( $k$ )	c	o	m	p	c	o	m	p	c
Encrypted text ( $c$ )	f	s	z	c	k	g	x	x	w

In the above example, the first letter is ‘d’, and the corresponding key letter is ‘c’. The encrypted text is ‘f’. This is because if the English letter,  $p_i$ , is ‘a’, the encrypted letter,  $c_i$ , is ‘c’. If  $p_i$  is ‘b’,  $c_i$  becomes ‘d’. If  $p_i$  is ‘c’,  $c_i$  becomes ‘e’. So, ‘d’ becomes ‘f’, if  $k_i$  is ‘c’.  $i = 0 \dots \text{len}(p) - 1$ . So, the relation is:

$$c_i = \text{str}((\text{ord}(p_i) + \text{ord}(k_i) - 2 * \text{ord}('a')) \bmod 26 + \text{ord}('a'))$$

Note that if  $p$  is longer than  $k$ ,  $k$  will be reused. Therefore,  $j = i \bmod \text{len}(k)$ . In the above example,  $k$  is thus “compcomp”.

Answer the following questions:

- Why does the above relation require “mod 26”? Illustrate with an example.
- Write down the *pseudo-code* to illustrate the above encryption process with the following input-output specifications:

Input:  $p$  and  $k$   
Output:  $c$

- Write down the *pseudo-code* to illustrate the decryption process, i.e., given  $c$  and  $k$ , convert  $c$  to  $p$ , with the following input-output specifications:

Input:  $c$  and  $k$   
Output:  $p$

- [25 marks] Suppose there is a “coin-moving game” and here is the description:
  - M square tiles are placed consecutively in a straight line. The distance between every two adjacent tiles is 1.
  - N coins are placed in N different tiles which may not be consecutive. You are asked to move all the piles to the same tile, which  $M \geq N$ .
  - Find the section which will require the smallest total moving distance.
  - Write down the pseudo-code of how to move all the coins to one single tile and count the total number of moves. Write down the *input* and *output* specifications.

b) Use your solution in 2a) as a function and write down the pseudo-code to find the smallest total moving distance. Write down the *input* and *output* specifications.

3. [30 marks] Complete the following tasks:

a) **Create** your own *min* function in Python, which finds the minimum number and its location (in zero-based index) in a set of different numbers.

Write a function, called `partA()`, to test your *min* function. When `partA()` is called, the input/output of your program will look like below:

```
Please enter a list of different numbers separated by ',': 4, -5, 6, 2, 0, 1, -7, 10, 3
The minimum number is -7.
Its location is 6.
```

b) Using your *min* function in 3a), implement a *sorting* function based on the method discussed in Lecture 4 to sort a set of different numbers. The function will return a list of sorted values in ascending order.

Write a function, called `partB()`, to test your *sort* function. When `partB()` is called, the input/output of your program will look like below:

```
Please enter a list of different numbers separated by ',': 4, -5, 6, 2, 0, 1, -7, 10, 3
A list of sorting values in ascending order: [-7, -5, 0, 1, 2, 3, 4, 6, 10].
```

You need to include *docstring* to describe the *min* and *sort* functions. Also, zero mark will be awarded if the built-in/external functions, *min* and *sort*, are used.

4. [20 marks] Develop a Python function, named `changeString()`, to update the character in a string in a particular location. It should accept a string, a character, and the index, as parameters and return the converted string.

Write a function, called `main()`, to test your `changeString()` function. When `main()` is called, the input/output of your program will look like below:

```
Input a string: Hello World
Input a zero-based location in a string to be changed: 5
Input a character to be updated in that location: @
The string updated: Hello@World
```

## Submission Instructions

Follow the steps below:

1. Create a folder and name it as <student no>\_<your name>, e.g., **12345678d\_CHANTaiMan**
2. For Q1 and Q2, type your answers in a word document and save it as a **.pdf** file. Name the single **.pdf** file as A2\_<student no>\_<your name> **.pdf**, e.g., **A2\_12345678d\_CHANTaiMan.pdf**
3. For Q3 and Q4, submit the source file (**.py**). Name the **.py** files as A2\_Q<question no>\_<student no>\_<your name> **.py**, e.g., **A2\_Q3\_12345678d\_CHANTaiMan.py**
4. Put all the **.pdf** and **.py** files into the folder created in Step 1.
5. Compress the folder (**.zip**, **.7z**, or **.rar**).
6. Submit the file to Blackboard.

A maximum of **3 attempts** for submission are allowed. **Only the last attempt will be graded.** A late penalty of 5% per hour will be imposed.

**Any wrong file naming and submission will be given ZERO mark.** If you are using Windows, the file extension may be hidden by the operating system. Follow the steps of below links to make sure the file extension is not hidden:

<https://www.howtohaven.com/system/show-file-extensions-in-windows-explorer.shtml>

**If your program cannot be run successfully (i.e., having any syntax error(s)), ZERO mark will be awarded for that program, regardless of how much you have coded.**

This assignment is an individual work. All work must be done on your own. Plagiarism is serious offence. You are not allowed to consult any external channels, e.g., discussion forums, and copy code from any web resources, to assist your completion of your assignments. The Moss (<https://theory.stanford.edu/~aiken/moss/>) system will be adopted for plagiarism checking for program code. Submissions with high similarity, in terms of code patterns and structures, in addition to direct-copy-and-paste, will be extracted and reviewed. Any plagiarism cases (both copier and copiee) will be given ZERO mark plus a deduction of the maximum mark of this assignment. Serious cases would be submitted to the Student Discipline Task Group (SDTG) of the department for further disciplinary actions.