

Singapore Management University  
School of Computing and Information Systems  
2024/2025 Semester 1  
IS628: Computational Thinking with Python  
Assignment 1 (due date: **6 Oct 2024, 11:59pm**)

**Maximum marks = 40.**

**Submission Instructions:**

You are expected to submit as following:

- (1) The skeleton code `a1.ipynb` is provided; read the comments, and fill in the missing parts;
- (2) Run the Feedback section in `a1.ipynb`;
- (3) Check `feedback.txt` in the same folder.
- (4) Zip your finalized `a1.ipynb` and `feedback.txt` then upload the zip file to the eLearn Assignment folder **Assignment 1**.

*\* Submissions in wrong format may not be run through auto-grader.*

Multiple submissions are permitted up to the due time, but only the last submission will be saved and graded. Please note that the Originality Check has been enabled.

**[Question 1] (5 marks)**

Train A departs from city  $P$  to city  $Q$ .  $x$  hours later, train B departs from city  $P$  to city  $Q$ . The speed of train A and the speed of train B are represented by  $V_A$  and  $V_B$  respectively with  $V_B > V_A$ . How many hours does it take for the two trains to meet?

Amy is explaining the above scenario to her ten-year-old daughter using toy blocks.

1. Initial set up: She uses  $V_A * x$  blocks to denote the location of train A after  $x$  hours.
2. She then begins to keep track of the time, for each hour, train A moves by  $V_A$  blocks while train B moves by  $V_B$  blocks.
3. The answer will be the hour count when train A and train B are in the same location.

**Sample input**

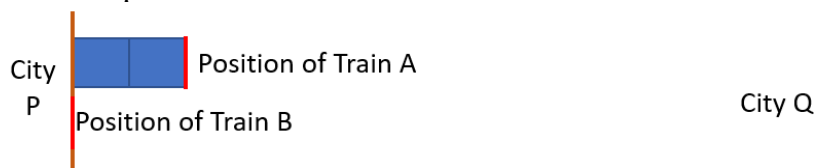
$$x = 2, V_A = 1, V_B = 2$$

**Expected output**

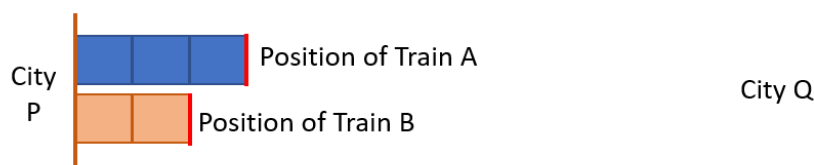
2

**Explanation**

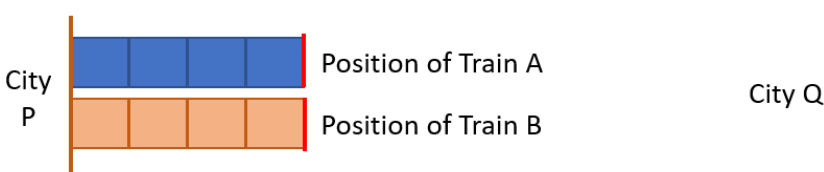
Initial set up:



After 1 hour:



After 2 hours:



Assumptions:

- The distance between city P and city Q is long enough for two trains to meet.
- All the numbers, including the answer, are integers.

Implement the above idea using **while loop**.

- 2 marks are allocated for sample case, which you can verify by running the 'Feedback' section in the provided `a1.ipynb`.
- The remaining 3 marks will be assessed after due date.

- 2 marks for hidden cases
- 1 mark will be awarded if you
  - Use a **while loop** to implement the above idea.

**[Question 2] (5 marks)**

A special kind of list contains only two elements, the first element is an integer, and the second element is either the python `None` value or a list of the same kind.

For example, `[3, [2, [1, None]]]` is one such list.

A function `eval()` takes a list of the above kind, sorts the integers in the list in ascending order, and returns a list in the same form.

An input list should follow the same format as the sample input below.

**Sample input**

```
[3, [2, [1, None]]]
```

**Expected output**

```
[1, [2, [3, None]]]
```

- 3 marks are allocated for sample case, which you can verify by running the 'Feedback' section in the provided `a1.ipynb`.
- The remaining 2 marks will be assessed for hidden case after due date.

**[Question 3] (12 marks)**

This game is to draw horizontal bars with the given list of words:

1. Sort the words by the alphabetical order of the second letter of each word then by the word length.
2. Assign each word a score that is equal to the length of the word minus 2.
3. Change the sign of the score obtained in step 2 to negative if a word starts with the letter 's'.
4. For each word with non-zero score, draw horizontal bars and one vertical bar in one row. The number of horizontal bars should be equal to the absolute value of the score. For a score less than zero, horizontal bars appear on the left side of the vertical bar, otherwise, horizontal bars appear on the right side of the vertical bar if the score is greater than zero. In addition, vertical bars are aligned vertically across rows.

**Sample input**

```
['it', 'set', 'seek', 'grade']
```

**Expected output**

The function `eval()` should return `'0 =|\n1==|\n2 |===\n3 | \n'`, when the return string is printed out, it looks like the plot shown below:

```
0 =|
1 ==|
2  |===
3  |
```

Assumptions:

- Words are in lowercase.
- Word lengths are greater than or equal to 2.
- No special characters in the word list.
- No two words share the same second letter and the same length.

**Explanation**

- According to step 1, words are sorted as `['set', 'seek', 'grade', 'it']`
  - In alphabetical order `'e' < 'r' < 't'`
  - Length of `'set'` < length of `'seek'`
- The words are assigned with scores `[1, 2, 3, 0]` at step 2.
- The scores will then be updated to `[-1, -2, 3, 0]` at step 3.
  - Both `'set'` and `'seek'` start with letter `'s'`
- At step 4, horizontal bars are drawn beside vertical bars which are aligned across rows. The absolute value of the score defines how long the bar is, and the sign of the score determines which side the horizontal bars appear.
  - The first score from step 3 is `-1`, one `'='` is drawn to the left of the vertical bar.
  - Similarly, the second score `-2` makes two `'='` to the left of the next vertical bar.
  - The third score makes three `'='` to the right of the next vertical bar.
  - No horizontal bars are drawn in the last row since the last score is zero, and only the vertical bar is drawn.

- 6 marks are allocated for sample case, which you can verify by running the 'Feedback' section in the provided `a1.ipynb`.
- The remaining 6 marks will be assessed after due date.
  - 4 marks for hidden cases
  - 2 marks will be awarded if you
    - Write a one-line lambda expression to do step 1.
    - Write a one-line list comprehension to do step 2 and step 3.

**[Question 4] (10 marks)**

A digital clock which displays time in the format of MM:DD:hh:mm (Month-Month, Day-Day, hour-hour in 24 hours format, and minute-minute) will occasionally show a string that looks the same backward as forward, which we call a palindromic clock time. For example, if it is 1220hrs on the day of Feb 21<sup>st</sup> 2024, the displayed clock time is 02:21:12:20. In this question, your task is to list all palindromic clock time in chronological order for a given interval defined by two calendar days.

The input is described by one line containing two specific dates separated by a space, and the dates are written in the format of “dd/mm/yyyy”. It is guaranteed (meaning there is no need to write code to check) that (i) the dates are valid, (2) the first date is earlier than the second date and (3) the duration between the two dates is not longer than 1 year. Print all palindromic clock time between the two days inclusive (meaning from the 0000hrs of the first date to 2359hrs of the second date) in the format of MM:DD:hh:mm where hh is in 24 hours format. Palindromic clock time should be arranged in chronological order, one per line.

**Sample Input**

10/12/2024 07/01/2025

**Sample Output**

```
12:10:01:21
12:11:11:21
12:12:21:21
12:20:02:21
12:21:12:21
12:22:22:21
12:30:03:21
12:31:13:21
01:01:10:10
```

- 2 marks are allocated for sample case, which you can verify by running the 'Feedback' section in the provided `a1.ipynb`.
- The remaining 8 marks will be assessed for hidden cases after due date.

**[Question 5] (8 marks)**

This question is to generate the dice image.

Write a function, which takes an integer ranging from 1 to 6, inclusively, and returns a string representing a text image of a dice of n.

For example, `eval(4)` will output a string `'o o\n \no o'`, if printed out, will look like this:

```
o o
o o
```

You can assume that n is always within the range 1 to 6 inclusive.

The six text images form are given to you in the following:

```
i = 1          i = 4
              o o
o
-----
i = 2          i = 5
o
o
-----
i = 3          i = 6
o
o
o
o o
o o
o o
```

<u>Sample Input</u>	<u>Expected Output</u>
1	' \n o \n '
2	' o\n \no '
3	' o\n o \no '
4	'o o\n \no o'
5	'o o\n o \no o'
6	'o o\nno o\nno o'

- 3 marks are allocated for sample case, which you can verify by running the 'Feedback' section in the provided `a1.ipynb`.
- The remaining 5 marks will be awarded after due date if you
  - Fill your code in only one line.
  - Use less than or equal to 85 characters to answer