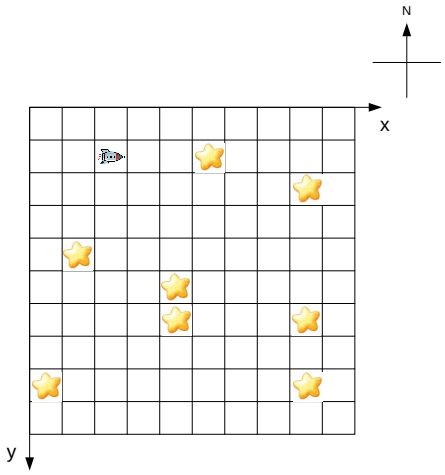
Computational Thinking and Problem Solving (COMP1002) and Problem Solving Methodology in Information Technology (COMP1001)

Assignment FIVE
(Due at noon on **20 November 2020**)

1. [20 marks] Write a Python program that generates the calendar of 2020, given that you only know that the first day of January is Wednesday and the number of days of each month. On input a month in its full name, the program displays the calendar of that month. The whole calendar cannot be stored in your program as variables and has to be generated dynamically based on the above information that you have. Assume user input is always correct and no data validation is required. Your program MUST look like the following with the same format:

```
Please enter a month in its full name or -1 to end: April
S   M   T   W   T   F   S
            1   2   3   4
5   6   7   8   9   10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30
Please enter a month in its full name or -1 to end: August
S   M   T   W   T   F   S
                        1
2   3   4   5   6   7   8
9   10  11  12  13  14  15
16  17  18  19  20  21  22
23  24  25  26  27  28  29
30  31
Please enter a month in its full name or -1 to end: November
S   M   T   W   T   F   S
1   2   3   4   5   6   7
8   9   10  11  12  13  14
15  16  17  18  19  20  21
22  23  24  25  26  27  28
29  30
Please enter a month in its full name or -1 to end: -1
Bye!
```

2. [30 marks] Modify A4 Q2. You are allowed to use the sample program in A4 (to be released after A4 submission deadline) or use your own version. Now Alpha can move around the space based on user's input. The user enters either N, E, S, or W, meaning the direction of the movement, and press "Enter". Alpha moves one step forward to the specified direction. If Alpha reaches a star, the program terminates. If the movement exceeds the boundary, it will remain in the current position. For each movement, the program will print out the updated map. Your program should have at least two newly defined functions, other than **`main()`**.



Your program should look like below:

```
  0 1 2 3 4 5 6 7 8 9
0       *   x
1               *
2       *
3     *
4
5 *           *
6
7       * *
8
9
Please enter the direction (N, E, S or W):S
  0 1 2 3 4 5 6 7 8 9
0       *
1           x   *
2       *
3     *
4
5 *           *
6
7       * *
8
9
Please enter the direction (N, E, S or W):W
  0 1 2 3 4 5 6 7 8 9
0       *
1         x     *
2       *
3     *
4
5 *           *
6
7       * *
8
9
```

```
Please enter the direction (N, E, S or W):N
  0 1 2 3 4 5 6 7 8 9
0       * x
1               *
2       *
3     *
4
5 *           *
6
7       * *
8
9
Please enter the direction (N, E, S or W):N
  0 1 2 3 4 5 6 7 8 9
0       * x
1               *
2       *
3     *
4
5 *           *
6
7       * *
8
9
Please enter the direction (N, E, S or W):W
  0 1 2 3 4 5 6 7 8 9
0
1               *
2       *
3     *
4
5 *           *
6
7       * *
8
9
Alpha crashes!
>>>
```

3. [25 marks] In Assignment 4, you are able to multiply multiple arbitrarily large numbers, but they are all integers. It has been suggested that you can multiply multiple arbitrarily large numbers that contain decimal points, based on the program for multiplying multiple arbitrarily large integers. The trick is to locate the decimal points from the list of integers and insert back the decimal point in the final answer at the proper position.

   (a) (10 marks) Develop a Python program that can multiply a list of large numbers which may be decimals. Use functions whenever appropriate. You can assume that all functions in Assignment 3 and Assignment 4 are available, i.e. you can make a "call" to those functions. You could make use of string functions, if needed. Provide appropriate comments in your program. Here are some sample outputs from your program:

```
>>> main(["61.431","1002"])
Input list ['61.431', '1002']
The product is 61553.862

>>> main(["61.431","10.02","61.434","1001","12345.6789","98765.4321"])
Input list ['61.431', '10.02', '61.434', '1001', '12345.6789', '98765.4321']
The product is 461549327683872 91.5640136294363052
```

   (b) (15 marks) A related challenge to multiplication is to do the reverse, i.e. division. One simple way to do a division N_1 / N_2 making use of multiplication is to estimate a and b, such that $a\ N_2 < N_1 < b\ N_2$ and gradually refine a (increasing a) and b (decreasing b) until a and b are close enough to each other. One key function that is useful to support a potential implementation of this "division" approach is a function **`compareNumber()`**.

   Assuming that x and y are large numbers, write the Python function **`compareNumber`**(x,y) that will return **`-1`** if $x < y$, return **`+1`** if $x > y$, and return **`0`** if $x = y$. Since we are not converting x and y into binary as in a computer, we can safely compare x and y for equality, unlike comparing floating numbers in a computer with a high risk (equality can hardly be true). Provide appropriate comments in your program. Explain briefly how you solve this problem via the use of *docstring* at the top. Here are some sample outputs from your function:

```
>>> print(compareNumber("61431","61434"))
Comparing 61431 61434
-1
>>> print(compareNumber("6143.1","614.34"))
Comparing 6143.1 614.34
1
>>> print(compareNumber("61431.1002","61434.1001"))
Comparing 61431.1002 61434.1001
-1
>>> print(compareNumber("12345678.987654321","12345678.9876543210"))
Comparing 12345678.987654321 12345678.9876543210
0
```

4. [25 marks] Consider the Pouring Water Puzzle.

   (a) (10 marks) Draw the graph showing the 16 nodes and the corresponding edges with respect to the original problem with a bowl of size 5-l and another bowl of size 3-l. From your graph, show that measuring out 4-l is the hardest problem among other similar problems with these two bowls.

   (b) (15 marks) There are 18 nodes when we consider the case with a bowl of size 5-l and another bowl of size 4-l and 20 nodes for a case with a bowl of size 7-l and another bowl of size 3-l, both more than the number of nodes in the original problem. It might appear that when the sizes of the two bowls increase, there are more nodes in the graph.

   Draw the graph showing the nodes and edges with a bowl of size 9-l and another bowl of size 12-l to show that this "conclusion" with more nodes for larger bowls is not necessarily correct. Are you able to measure out 4-l with these two bowls? What volume could be measured out? Explain briefly on the observation.

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, Q2 and Q3. You need to submit the source file (**`.py`**). Name the **`.py`** files as A5_Q<question no>_<student no>_<your name>**`.py`**,
   e.g., **`A5_Q1_12345678d_CHANTaiMan.py`**
3. For Q4, you need to put your answers in a word document and/or take the image from some tool and save it as a **`.pdf`** file. Alternatively, you could draw on a piece of paper and scan the answers into a **`.pdf`** file. Name the single **`.pdf`** file as A5_<student no>_<your name>**`.pdf`**,
   e.g., **`A5_12345678d_CHANTaiMan.pdf`**
4. Put all the **`.py`** and **`.pdf`** files into the folder.
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.