FIT1045 Algorithmic Problem Solving – Assignment 1 (5%).

Due: 11:55 pm, Sunday 2nd September, 2018.

Objectives

The **objectives of this assignment** are:

- To demonstrate the ability to implement algorithms using basic data structures and operations on them.
- To gain experience in designing an algorithm for a given problem description and implementing that algorithm in Python.

Submission Procedure

1. Put you name and student ID on each page of your solution.
2. Save your files into a zip file called yourFirstName_yourLastName.zip
3. Submit your zip file containing your solution to Moodle.
4. Your assignment will not be accepted unless it is a readable zip file.

Important Note: Please ensure that you have read and understood the university's policies on plagiarism and collusion available at `http://www.monash.edu.au/students/policies/academic-integrity.html`. You will be required to agree to these policies when you submit your assignment.

Marks: This assignment has a total of 25 marks and contributes to 5% of your final mark. Late submission will have 5% off the total assignment marks per day (including weekends) deducted from your assignment mark. (In the case of Assignment 1, this means that a late assignment will lose 1.25 marks for each day (including weekends)). Assignments submitted 7 days after the due date will normally not be accepted.

Marking Guide:

Task 1: 10 marks

(a) Code readability (Non-trivial comments where necessary and meaningful variable names) – 2 marks

(b) Correct input handling – 2 marks

(c) Correct result – 2 marks

(d) Loop calculates each term correctly – 4 marks

Task 2: 15 marks

(a) Code readability (Non-trivial comments where necessary and meaningful variable names) – 4 marks

(b) Correct input handling – 4 marks

(c) Checking magic square property – 2 marks

(d) Loop allowing for user input – 2 marks

(e) Checking cell contains 0 – 1 mark

(f) Undoing if value not valid – 2 marks

Task 1: Continued Fractions 10 Marks

Write a Python program that takes as input non-negative integer n and then computes an approximation for the value of e using the first $n+1$ terms of the continued fraction:

$$e \approx 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{2}{3 + \cfrac{3}{\ddots}}}}$$

For example: If you entered 2, your program would output 2.7272727272727275 as:

$$e \approx 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{2}{3}}}$$

and if you entered 3, your program would output 2.7169811320754715 as:

$$e \approx 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{2}{3 + \cfrac{3}{4}}}}.$$

Task 2: Magic Squares 15 Marks

Information:
A magic square is a table with n rows and n columns, such that the numbers in each row, and in each column, and the numbers in the main diagonals, all add up to the same number which we will refer to as the *magic sum*. All entries are distinct. A *partial magic square* is a magic square that has some entries missing. For example, Table 1 gives an 4×4 magic square and Table 2 gives a partial magic square.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Table 1: Magic Square (magic sum is 34).

16		2	13
5	10	11	8
9	6		12
	15	14	1

Table 2: Partial Magic Square (magic sum is 34).

Write a Python program that takes as input the name of a file containing a potential partial magic square and a number magicSum, the magic sum for that square. Each line of the file is a row in the magic square. Cells containing blank entries are denoted by a 0.

For example: the following input file would represent the partial magic square given in Table 2.
16 0 2 13
5 10 11 8
9 6 0 12
0 15 14 1

Your program should check that the magic square is a partial magic square, that is, it has the following property:

Magic Square Property The sum of every row, column and the main diagonals is **at most** the number magicSum.

Your program should repeatedly do the following until the user elects to stop.

1. Print the table representing the partial magic square.
2. Ask the user to select a cell in the table and a number to put into the cell.

3. Your program should then respond in one of the following ways:

   - If the cell already has a non-zero element, inform the user that this is an invalid entry.
   - Insert the number into this cell. Check if the updated table violates the magic square property. If it does, inform the user that this is an invalid entry and put 0 back into that cell.

Some Partial Magic Squares:

14	10	1		18
20	11			24
21	17	13		
	23	19		6
8	4			12

Table 3: Magic sum is 65.

68		16	40	44
96			48	72
		52		
32			84	
60	64		12	36

Table 4: Magic sum is 260.