Assignment 1 Questions

This document only contains the questions for Assignment 1. Please refer to the Assignment 1 rubric file in Canvas for instructions on how to submit the assignment.

Question 1

Write a program to read a text file `q1_in.txt`. The first line will contain a number n s.t. ($1 < n < 100$). The following n lines will contain n numbers $i_1, i_2, \ldots, i_n$.

```
n
i_1
i_2
...
i_n
```

Example Input File1:

```
5
1
2
3
4
5
```

Write a program to compute the Lucas number for each i_k for $k \in [1, n]$. The Lucas number L_i is defined as follows:

$L_1 = 2$

$L_2 = 1$

$L_i = L_{i-1} + L_i(i-2)$, for $i > 2$

Having created a list of Lucas numbers, then reverse the list, and print it on to a text file `q1_out.txt`.

```
L_{i_n}
L_{i_n-1}
...
L_{i_2}
L_{i_1}
```

Example Output File1: (Corresponds to Example Input File 1)

```
7
4
3
1
2
```

Your output file should contain exactly $n + 1$ lines (inclusive of an empty line at the end).

Explanation of the example input/output:

The first line of the input file describes the number of integers to follow (5). The next 5 numbers tell us the Lucas numbers to calculate (1, 2, 3, 4, 5).

The Lucas numbers corresponding to these numbers are ($L_1 = 2, L_2 = 1, L_3 = 3, L_4 = 4, L_5 = 7$). We take the list of Lucas numbers (2, 1, 3, 4, 7), and reverse it to get (7, 4, 3, 1, 2). Then these numbers are printed on to a file, one number per line, followed by an empty line.

Example Input2:

```
2
2
5
```

Example Output 2:

```
7
1
```

Question 2

We want to make a reverse polish calculator to perform simple arithmatic operations of *addition (`+`), subtraction (`-`), integer division (`/`), and multiplication(`*`).

In reverse polish notation (RPN), operands appear prior to the operator. Eg:

- `4 5 +` in RPN stands for `4+5` .
- `30 3 /` stands for `30/3` .

You can even nest operations. Eg:

- `1 2 3 + -` stands for `1 - (2 + 3)`
- `10 4 9 * + 3 -` stands for `( 10 + (4 * 9) ) - 3`

- Your program should read a text file `q2_in.txt`.
  - The first line will contain a single string with an arithmatic computation in reverse polish notation.
- Your program should write to a text file `q2_out.txt`.
  - There should be a single line containing a single integer, the answer to the arithmatic computation given in the input.

Example Input File 1:

```
1 2 3 + -
```

Example Output File 1:

```
-4
```

Example Input File 2:

```
10 4 9 * + 3 -
```

Example Output File 2:

```
43
```

Question 3

Imagine you are a knight on a chess-board. (Consider a standard chess board and standard knight movements on a chessboard). You are given a starting location, and a destination. Write a program to output the smallest number of steps it takes for you to get to your destination.

- Your program should read a text file `q3_in.txt`. There will be two lines.
  - **First** line will have two characters describing the **starting location**.
  - **Second** line will have two characters describing the **destination**.
  - Locations will be in standard chess notation. (i.e. a1, b2, e5, h8 etc.)
- Your program should write to a text file `q3_out.txt`.
  - There should be a single line containing a single integer, the smallest number of steps required.

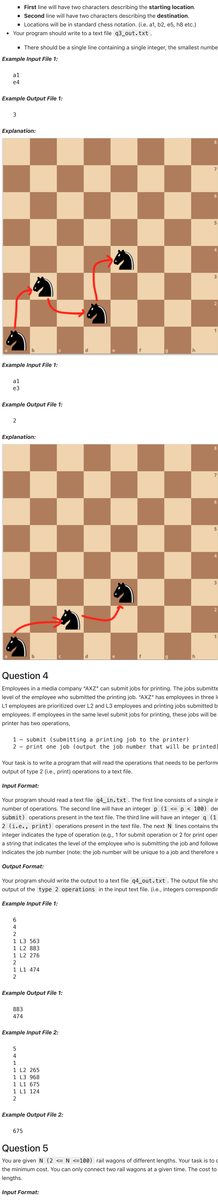
Example Input File 1:

```
a1
e4
```

Example Output File 1:

```
3
```

Explanation:



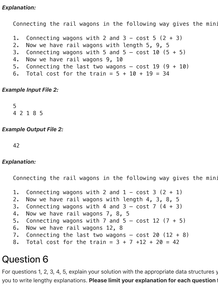
Example Input File 1:

```
a1
e3
```

Example Output File 1:

```
2
```

Explanation:



Question 4

Employees in a media company "AXZ" can submit jobs for printing. The jobs submitted for printing will be prioritized based on the level of the employee who submitted the printing job. "AXZ" has employees in three levels: `L1`, `L2`, `L3` . Printing jobs submitted by L1 employees are prioritized over L2 and L3 employees and printing jobs submitted by L2 employees are prioritized over L3 employees. If employees in the same level submit jobs for printing, these jobs will be processed in the order it was submitted. The printer has two operations,

```
1 - submit (submitting a printing job to the printer)
2 - print one job (output the job number that will be printed),
```

Your task is to write a program that will read the operations that needs to be performed by the printer from a text file and write the output of type 2 (i.e., print) operations to a text file.

Input Format:

Your program should read a text file `q4_in.txt` . The first line consists of a single integer `N` (`1 <= N <=100`) denoting the number of operations. The second line will have an integer `p` (`1 <= p <= 100`) denoting the number of `type 1 (i.e., submit)` operations present in the text file. The third line will have an integer `q` (`1 <= q < 100`) denoting the number of `type 2 (i.e., print)` operations present in the text file. The next `N` lines contains the details of the operations. In each line, the first integer indicates the type of operation (e.g., 1 for submit operation or 2 for print operation). If the `type is 1` , it is first followed by a string that indicates the level of the employee who is submitting the job and followed by an integer `r` (`1 <= r <=1000`) that indicates the job number (note: the job number will be unique to a job and therefore will not be repeated).

Output Format:

Your program should write the output to a text file `q4_out.txt` . The output file should have exactly `q` lines corresponding to the output of the `type 2 operations` in the input text file. (i.e., integers corresponding to the job number of the print operations)

Example Input File 1:

```
6
4
2
1 L3 563
1 L2 883
1 L2 276
2
1 L1 474
2
```

Example Output File 1:

```
883
474
```

Example Input File 2:

```
5
4
1
1 L2 265
1 L3 968
1 L1 675
1 L1 124
2
```

Example Output File 2:

```
675
```

Question 5

You are given `N` (`2 <= N <=100`) rail wagons of different lengths. Your task is to connect all these wagons to create a train with the minimum cost. You can only connect two rail wagons at a given time. The cost to connect two wagons is equal to the sum of their lengths.

Input Format:

Your program should read a text file `q5_in.txt` . The first line consists of a single integer `N` (`2 <=N <=100`) denoting the number of rail wagons. The second line will contain integers `r1 r2 ... rN` corresponding to the length of each rail wagon.

Output Format:

Your program should write the output to a text file `q5_out.txt` . The output file should contain an integer in a single line corresponding to the minimum cost to connect these wagons.

Example Input File 1:

```
4
9 2 3 5
```

Example Output File 1:

```
34
```

Explanation:

```
Connecting the rail wagons in the following way gives the minimum cost train.

1.  Connecting wagons with 2 and 3 - cost 5 (2 + 3)
2.  Now we have rail wagons with length 5, 9, 5
3.  Connecting wagons with 5 and 5 - cost 10 (5 + 5)
4.  Now we have rail wagons 9, 10
5.  Connecting the last two wagons - cost 19 (9 + 10)
6.  Total cost for the train = 5 + 10 + 19 = 34
```

Example Input File 2:

```
5
4 2 1 8 5
```

Example Output File 2:

```
42
```

Explanation:

```
Connecting the rail wagons in the following way gives the minimum cost train.

1.  Connecting wagons with 2 and 1 - cost 3 (2 + 1)
2.  Now we have rail wagons with length 4, 3, 8, 5
3.  Connecting wagons with 4 and 3 - cost 7 (4 + 3)
4.  Now we have rail wagons 7, 8, 5
5.  Connecting wagons with 7 and 5 - cost 12 (7 + 5)
6.  Now we have rail wagons 12, 8
7.  Connecting the last two wagons - cost 20 (12 + 8)
8.  Total cost for the train = 3 + 7 +12 + 20 = 42
```

Question 6

For questions 1, 2, 3, 4, 5, explain your solution with the appropriate data structures you used to solve the problem. We don't expect you to write lengthy explanations. **Please limit your explanation for each question for a maximum of 200 words.**