LAB EXAM (15 February 2022 : 1400-1550)

Before starting

- The exam runs 1400-1550.
- Answer all questions.
- Upload all code to Einstein.
- All lab exam rules apply.

Question 1 [25 marks]

- Write a program called `unique_061.py` that reads one or more lines of text from `stdin`.
- Each line consists of one or more integers.
- Your program should output the largest unique integer in each line or *none* if no such value exists. (An integer is unique if it occurs only once.)
- For example:

```
$ cat unique_stdin_00_061.txt
1 1 1 5 3 4 6 6
4 4 4
```

```
$ python3 unique_061.py < unique_stdin_00_061.txt
5
none
```

- Hint: You may find the `count()` method useful. For example:

```
alist = [1, 2, 3, 4, 5, 2]
print(alist.count(3))
print(alist.count(2))
```

```
1
2
```

Question 2 (part 1) [15 marks]

- Write a program called `nice_061.py` that reads a list of words from `stdin` (one word per line). Each word consists of lower case letters.
- Your program must print each word in the list that contains *just a single instance* of each of the characters *n*, *i*, *c*, *e*.
- Words must be output in the order they are read from `stdin`.
- For example:

```
$ cat nice_stdin_00_061.txt
angelic
novice
panicked
unpanicked
arsenic
jaundice
```

```
$ python3 nice_061.py < nice_stdin_00_061.txt
angelic
novice
panicked
arsenic
jaundice
```

Question 2 (part 2) [10 marks]

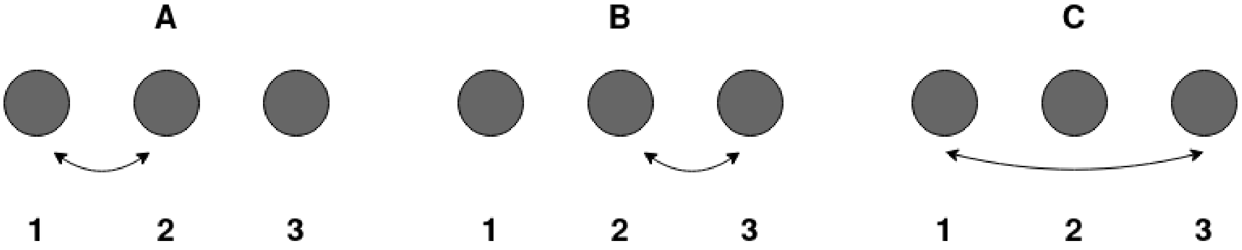
- Write a program called *`order_061.py`* that reads a list of words from `stdin` (one word per line). Each word consists of lower case letters.
- Your program must print each word in the list that meets two conditions: Firstly, the word must contain *just a single instance* of each of the characters *n*, *i*, *c*, *e*, and, secondly, those characters must occur in the order *n* before *i* before *c* before *e*.
- Words must be output in the order they are read from `stdin`.
- For example:

```
$ cat order_stdin_00_061.txt
angelic
novice
panicked
unpanicked
arsenic
jaundice
```

```
$ python3 order_061.py < order_stdin_00_061.txt
novice
panicked
jaundice
```

Question 3 [25 marks]

- We have three cups. They can be swapped as shown below.



- A prize is placed under a cup to start with and then a series of swaps are carried out.
- Write a program called *`prize_061.py`* that reads two lines of text from `stdin`.
- The first line is an integer in the range 1-3 indicating the starting position of the prize.

- The second line consists of a string (of minimum length one) over the characters *A*, *B*, *C* denoting a sequence of swaps to be carried out.
- Your program should output the final location of the prize.
- For example:

```
$ cat prize_stdin_00_061.txt
1
AB
```

```
$ python3 prize_061.py < prize_stdin_00_061.txt
3
```

- For example:

```
$ cat prize_stdin_01_061.txt
2
CBBA
```

```
$ python3 prize_061.py < prize_stdin_01_061.txt
1
```

Question 4 [25 marks]

- Write a program called *`sales_061.py`* that reads one or more lines of sales data from `stdin`.
- Each line consists of a salesperon’s name followed by one or more comma-separated sales figures for that salesperson and is structured as follows:

  Salesperson name: sale, sale, …, sale

- A salesperson’s overall sales average is the sum of their individual sales figures divided by their number of sales.
- You can assume all names are unique and all sales figures are positive integers.
- Having read in all lines the program should print a results table in descending order of overall average sales. Average sales values must be printed to two decimal places.
- You can assume there will be no ties.
- For example:

```
$ cat sales_stdin_00_061.txt
Sheila: 5, 8, 2
Miranda: 1, 2, 1
Jenny: 2, 3
Irene The Invincible: 1200
Wendy O'Brien: 10, 2
```

```
$ python3 sales_061.py < sales_stdin_00_061.txt
Irene The Invincible: 1200.00
Wendy O'Brien: 6.00
Sheila: 5.00
```

```
Jenny: 2.50
Miranda: 1.33
```

- Hint: You may find it useful to `split()` strings on characters other than white space.