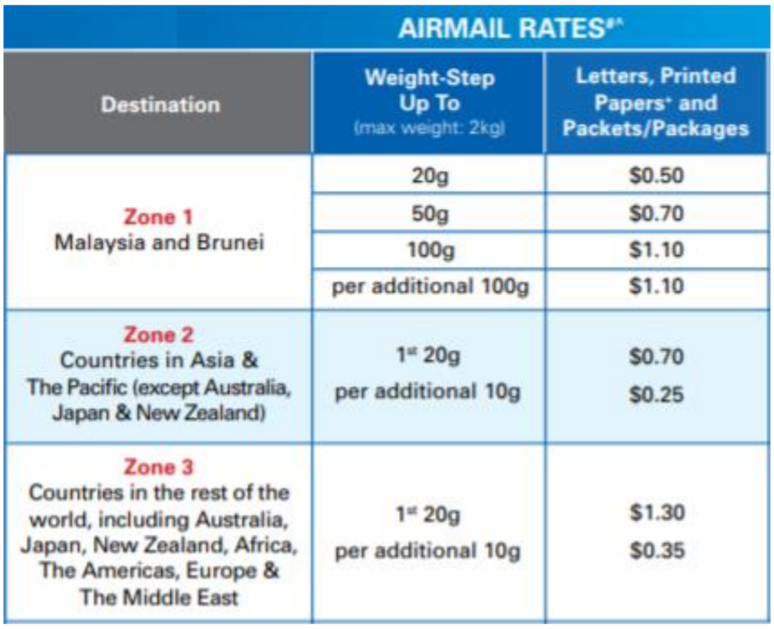
Lab 6 (1) – Revision

1. Write a program that compute the cost of delivering an item to a country, according to the table found in
   `https://www.singpost.com/sites/default/files/PostageRates-MailingGuidelines_052017.pdf`

   AIRMAIL RATES[#^]

Destination	Weight-Step Up To (max weight: 2kg)	Letters, Printed Papers[+] and Packets/Packages
Zone 1 Malaysia and Brunei	20g	$0.50
	50g	$0.70
	100g	$1.10
	per additional 100g	$1.10
Zone 2 Countries in Asia & The Pacific (except Australia, Japan & New Zealand)	1st 20g	$0.70
	per additional 10g	$0.25
Zone 3 Countries in the rest of the world, including Australia, Japan, New Zealand, Africa, The Americas, Europe & The Middle East	1st 20g	$1.30
	per additional 10g	$0.35

   The cost is dependent on the weight of the item and the zone of the destination. The program reads the weight and the zone and outputs the cost of postage. Weight must be positive and non-zero and zone must be either 1, 2 or 3.

2. Modify Question 1 so that a user may repeatedly enter the weight of the item and the zone of the destination for as many items as he wishes to send. Assume that data entry is terminated when user enters an empty string when prompted for the weight of item.

   At the end the data entry, output
   - the number of items and postage cost per zone, and
   - the total number and postage cost for all items

   Save the output into a file called Q2Out.dat

3. Manage the food stock of a small food store that sells 5 types of soup (Clam Chowder, Mushroom, Tomato, Pumpkin and Oxtail). Each type of soup starts with the quantity of 50 servings.

   Write a program that allows a user to choose one of the following options:
   ```
   1. Purchase Soup
   2. Replenish Soup
   0. Exit
   ```

Choosing option 1 will display only available types of soups (if there are servings left), in alphabetical order and quantity available. The user can choose the type of soup (identified by the first letter) and enter the servings required. The servings left for each soup should not be negative.

Option 2 allows a type of soup to be replenished. The user can enter the type of soup and quantity to replenish by. The final servings left for a soup that is replenished should not exceed 50.

4. Let multiple players play the high-low guessing game. At the start of the game session, ask for the names of players playing the guessing. Player names should not be repeated.

   For each game, track the winner(s) so that a summary of the winners for each game can be printed at the end of the game session.

   Example output:

```
Enter player's name or <Enter> to end: tom
Enter player's name or <Enter> to end: jane
Enter player's name or <Enter> to end: alex
Enter player's name or <Enter> to end:
Tom, make a guess:
Enter a guess between 1 and 100 inclusive: 50
Jane, make a guess:
Enter a guess between 1 and 100 inclusive: 75
Alex, make a guess:
Enter a guess between 1 and 100 inclusive: 25
Processing guess of Tom: 50 Too low.
Processing guess of Jane: 75 Too low.
Processing guess of Alex: 25 Too low.
Tom, make a guess:
Enter a guess between 1 and 100 inclusive: 80
Jane, make a guess:
Enter a guess between 1 and 100 inclusive: 85
Alex, make a guess:
Enter a guess between 1 and 100 inclusive: 90
Processing guess of Tom: 80 Too low.
Processing guess of Jane: 85 Too low.
Processing guess of Alex: 90 Too high
Tom, make a guess:
Enter a guess between 1 and 100 inclusive: 86
Jane, make a guess:
Enter a guess between 1 and 100 inclusive: 86
Alex, make a guess:
Enter a guess between 1 and 100 inclusive: 87
Processing guess of Tom: 86 Correct! in 3 tries
Processing guess of Jane: 86 Correct! in 3 tries
Processing guess of Alex: 87 Too high
Winners: Tom Jane
Continue game?(y/n): y
Tom, make a guess:
Enter a guess between 1 and 100 inclusive: 60
```

```
Jane, make a guess:
Enter a guess between 1 and 100 inclusive: 40
Alex, make a guess:
Enter a guess between 1 and 100 inclusive: 20
Processing guess of Tom: 60 Too high
Processing guess of Jane: 40 Too high
Processing guess of Alex: 20 Too high
Tom, make a guess:
Enter a guess between 1 and 100 inclusive: 10
Jane, make a guess:
Enter a guess between 1 and 100 inclusive: 13
Alex, make a guess:
Enter a guess between 1 and 100 inclusive: 15
Processing guess of Tom: 10 Too low.
Processing guess of Jane: 13 Too low.
Processing guess of Alex: 15 Correct! in 2 tries
Winner: Alex
Continue game?(y/n): n
Games Winners
Game 1 Winners: Tom Jane
Game 2 Winner: Alex
```