Lab 4 (File handling, Lists)

1. (File handling) The rainfall (in mm) for each day in a particular month is captured in a file, rainfall.dat. Write a program to read the rainfall for each day, and display a summary report as follows:

```
Rainfall Summary
Average rainfall 5.65mm
No of days with no rain 3 days
Highest rainfall recorded 20.6mm
```

   Refer to the Appendix of this document for the content of rainfall.dat which you can use to create the input file.

2. (File handling) The name, weight and height of some customers in a gym are stored in a comma separated file, customer.dat

a. Write a program that reads each line of customer.dat and as each line is read, it computes the body mass index (bmi) report as follows:

```
Name  Weight Height BMI
John  50     1.4    25.51
Peter 60     1.7    20.76
…
Joe   45     1.45   21.40
```

   BMI is computed using the formula: $bmi = \frac{weight}{height^2}$

   Refer to the Appendix of this document for the content of customer.dat which you can use to create the input file.

b. Modify the above program to store the report in an output file, bmi.dat

3. (List) Write a program to record the statistics of 100 throws of a dice.

   Use an initial list as follows to record the occurrences of each value of the dice:
```
diceCount = [0, 0, 0, 0, 0, 0, 0]
```

   The element at index 0 of the list will not be used. When the dice value is 1, add 1 to index 1 of the list etc. After 100 throws of the dice, display a summary of the occurrences of the dice values as follows:

```
Dice  Occurrence
1     16
2     14
3     15
4     17
5     18
6     20
Total 100
```

4. (String, tuple, list) A check digit is usually appended to a number in order to detect errors arising when the number is transcribed manually.

The check digit for the NRIC No. is the official reference which is determined as follows:

For example : NRIC No.(with official reference) = S7928964G

Step 1:
Multiply each digit by the following weights.

NRIC No. :	7	9	2	8	9	6	4
Weights :	**2**	**7**	**6**	**5**	**4**	**3**	**2**
Products:	14	63	12	40	36	18	8

Make use of a tuple for the weights:
```
weight = (2, 7, 6, 5, 4, 3, 2)
```

Step 2:
Sum the products of each digit x weight.
Sum : 14 + 63 + 12 + 40 + 36 + 18 + 8 = 191

Step 3:
Find the remainder when the sum is divided by 11.
Sum/11: 17 remainder 4

Step 4:
Take 11 – remainder to get the check digit.
Check digit : 11 – 4 = 7

Step 5:
Look up the following table to get the official reference.

Official Reference : G

Conversion:	A	B	C	D	E	F	G	H	I	Z	J
Table	1	2	3	4	5	6	7	8	9	10	11

Write a program that reads in a nric as a string and checks whether the reference is correct. Make use of a tuple for step 5, instead of using if..elif statement.

5.   (tuple, list) A test consists of 10 MCQ questions. Each question has 4 choices: a, b, c, d. The solution to each question is stored in a tuple as follows:
        ('a', 'b', 'b', 'a', 'd', 'c', 'b', 'a', 'b', 'c')

to indicate that the answer to question 1 is a, to question 2 is b, etc.

Write a program that allows a user to take the quiz. The user is prompted for his answers to the 10 MCQ questions. Assume input will be one of the 4 valid choices is entered: a, b, c, d. Store the answers in another collection.

After all the questions have been answered, the program displays if each of the questions is answered correctly, and if not, the program displays the correct answer. A summary of the number of correct answers is also displayed.

An example run is shown here:
        Q1: a

```
Q2: b
Q3: c
Q4: a
Q5: d
Q6: c
Q7: b
Q8: a
Q9: d
Q10: c
Q1: a correct
Q2: b correct
Q3: c incorrect, answer is b
Q4: a correct
Q5: d correct
Q6: c correct
Q7: b correct
Q8: a correct
Q9: b correct
Q10: c correct
Total 9 out of 10 correct
```

6. (Multiple lists) A program is required to record and display the results of a swimming competition. The swimming pool has 5 lanes. The swimmer name for each lane is first recorded. After the race, the timings for the swimmers are captured. The results for the race is then displayed.

a. Write a function `inputSwimmers()` that prompts for 5 swimmer names and returns the names in a list.

An example of the input session is as follows:
```
Enter lane 1 swimmer: James
Enter lane 2 swimmer: Joseph
…
Enter lane 5 swimmer: Ian
```
Test the function.

b. Write another function `inputTiming(swimmers)` that has the list of swimmers names. The function creates an empty list, and records the timing for each of the swimmers. The timings list is returned. An example of the input session is as follows:
```
Enter timing for James: 47.15
Enter timing for Joseph: 46.91
….
Enter timing for Ian: 48.01
```

c. Write another function `printResults(swimmers, timings)` that has the name and timing lists as parameters. The function prints a summary of the results as follows:
```
James        47.15s
Joseph       46.91s
…
Ian          48.01s
Fastest is 46.91s
```

Assume all the timings are unique. Write a main function to test out all the parts in this question.

d. Modify the function in part c) that prints the summary in sorted order of the timing. Therefore, the summary may look like this:

```
Joseph      46.91s
James       47.15s
…
Ian         48.01s
```

Appendix

Data files for Q1 and Q2
rainfall.dat
```
0.2
7.8
0.4
3.4
0.4
3.8
12
5.4
1.6
0
0.8
12.4
2.4
2
4.6
0.8
18.4
7.4
20.6
4
13.2
2
4
0
4.8
14.4
9.6
0
5.6
7.6
```

customers.dat
```
John,50,1.4
Peter,60,1.7
Amy,40,1.3
Nathan,70, 1.7
Joe,45,1.45
```