

EXAMINATIONS – 2020

TRIMESTER 2

COMP 132 Term Test
Programming for the Natural
and Social Sciences

Time Allowed: 2 hours

OPEN BOOK

Permitted materials: All materials, calculators and dictionaries are permitted

Instructions: Do not spend too much time on a question if you get stuck. For example, if you have to Google for a question, do it later after you work on other questions.

Write all your answers in the template file called `TermTest.ipynb`. Write your name and student ID at the top of your file as comments. Do not change the file name.

Save your file frequently to avoid data loss.

Submit your `TermTest.ipynb` file using our online submission system immediately after the test.

The term test will be marked out of 50.

Questions

1. Using if [7]
2. Writing loops [3]
3. Using lists [8]
4. Using Numpy arrays [12]
5. Data visualization [8]
6. Using strings [12]

Question 1. Using if**[7 marks]**

Define a function to calculate the delivery fee of a courier. The parameter of this function is the weight of an item, and the returned value is the delivery fee of the item. The delivery fee should be calculated as follows:

- If the weight is less than 20kg, the delivery fee is \$15.
- If the weight is in between 20kg and 50kg (include 20 and 50), the delivery fee is \$1 per kilogram.
- If the weight is greater than 50kg, the delivery fee is \$1.2 per kilogram plus an extra charge of \$10.

Test your function three times using different values for the weight.

Question 2. Writing loops**[3 marks]**

Write a `for` loop to print the numbers and their squares from 10 to 30 (include 10 and 30). Print the results in the following format:

```
10's square is 100
11's square is 121
12's square is 144
13's square is 169
...
30's square is 600
```

Question 3. Using lists**[8 marks]**

The following list `JVL_Johnsonville` contains station names of the Wellington train line JVL (Johnsonville-Wellington), and it is given in the template file `TermTest.ipynb`. Use the list to answer this question.

```
JVL_Johnsonville=['Johnsonville', 'Raroa', 'Khandallah', 'BoxHill',
                  'SimlaCrescent', 'AwaruaStreet', 'CroftonDowns',
                  'Wellington']
```

- (a) **[2 marks]** Print the third station of the JVL train line. Johnsonville is the first station.
- (b) **[2 marks]** Insert the station "Nagio" between `AwaruaStreet` and `CroftonDowns`, and print the list.
- (c) **[4 marks]** Use this list to create a new list named `JVL_Wellington` which contains these stations in reverse order, and print it out. Your code should not change the `JVL_Johnsonville` list. Print `JVL_Johnsonville` at the end of your code to show that it is not changed.

Question 4. Using Numpy arrays**[12 marks]**

The Numpy library is imported at the beginning of this question. The template file also gives you two lists of numbers in the comments. Suppose the data is the number of Flamingo scooters rented in each hour (from 0 to 23) in two days. Your code should still work if the numbers are changed.

```
import numpy as np
#day 1 data: 2, 4, 9, 0, 1, 13, 6, 17, 31, 16, 9, 10, 13, 13, 14, 14, 23,
45, 38, 28, 16, 15, 9, 5
#day 2 data: 3, 1, 16, 7, 6, 16, 54, 16, 29, 16, 10, 12, 18, 16, 14, 17,
20, 42, 36, 28, 16, 15, 9, 4
```

Write code to complete the tasks below:

- (a) **[2 marks]** Create two **Numpy arrays** `day1` and `day2` containing the numbers.
- (b) **[2 marks]** Find and print the total number of scooters rented in `day1` and `day2`.
- (c) **[2 marks]** Find out in which hours the rented scooters are more than 20 in `day1` . Print out the number of the rented scooters and the corresponding hours.
- (d) **[4 marks]** Suppose the early morning hours are from 6am to 10am, and late afternoon hours are from 4pm to 8pm. Write code to find whether the scooters are more rented in early morning hours or late afternoon hours in `day1`. Print `Early morning is busy` or `Late afternoon is busy` accordingly.
- (e) **[2 marks]** Find the peak hour in `day1` and print the hour and the number of scooters rented in this hour. The peak hour is the hour when the most scooters are rented.

Question 5. Data visualisation**[8 marks]**

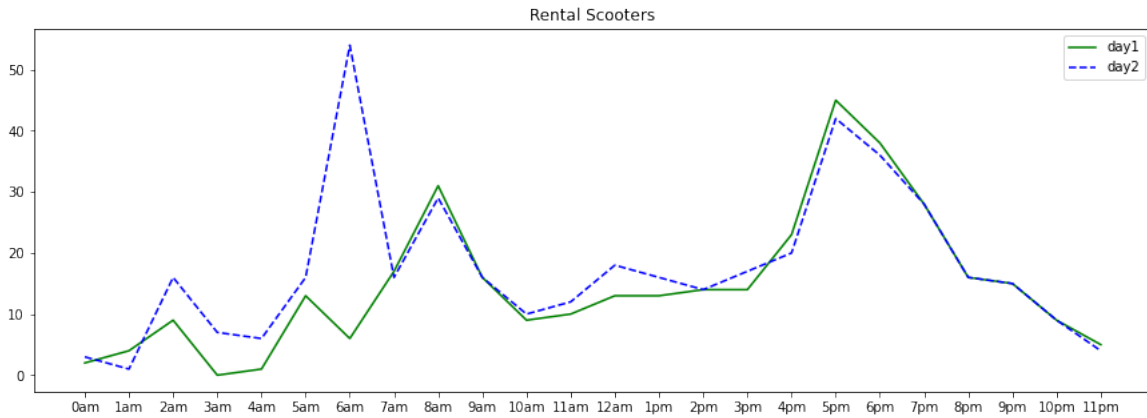
The template file has set up your program as follows:

```
%matplotlib inline
import matplotlib.pyplot as plt
```

Using the same data as in Question 4, write code to visualize the data:

- (a) **[2 marks]** Plot the data of the two days using different color and line styles.
- (b) **[1 mark]** Add a title to the figure.
- (c) **[1 mark]** Add a legend to distinguish the data for two days.
- (d) **[4 marks]** Make the horizontal axis display hours with am and pm at the proper positions. Find a way to make the figure bigger so that the hour labels are not overlapping.

Your figure should look like this:



Question 6. Using strings

[12 marks]

Use the string given in `gene` as an example to answer this question.

```
gene="GGCTCTTTAGCTTCTTAGGACTGCAAGCTAACGTTGCTTTGTTTTCAACTTCTAATCCTTTGAGTGTTTTCA"
```

Your code should still work if the string is changed to a similar gene string.

- (a) **[4 marks]** Find the most frequent letter in the string. Print this letter and its frequency. You may assume that there are only four kinds of letters ('T', 'A', 'G', 'C') in all gene strings.
- (b) **[4 marks]** Find all occurrences of the pattern 'AG' and print their starting indexes. For the example data given in `gene`, it should print 8, 16, 25, 62.
- (c) **[4 marks]** Divide the string into groups of 10 letters, print each group in a new line. For the example given in `gene`, your program should print:

```
GGCTCTTTAG
CTTCTTAGGA
CTGCAAGCTA
ACGTTGCTTT
GTTTTCAACT
TCTAATCCTT
TGAGTGTTTT
CA
```
