**Raffles Institution**

**Year 2 Computer Elective Programme**

Summative 2

Batch-Processing Medium Sized Data

Weightage: 15%

**Building Maintenance**

You are a new employee of the building maintenance department at NUS and you have just been given one data file timetable.csv in CSV (comma-separated values) format that contains timetabling information about NUS modules for one week. Your boss wants you to do some analysis of the building occupancy for the university.

You are provided with two smaller sets of timetable.csv files and the following is the expected results for the two sets of files. Note that your code will be tested with other data files with the same input format, so just because your code works with the files provided does not necessarily mean that your code is correct.

**A.** You are to write a function total­\_venues that takes as input a file and returns the number of different venues that are used by the various modules.

Sample execution:

>>> total\_venues("timetable1.csv")

344

>>> total\_venues("timetable2.csv")

71

**B.** You are supposed to help with enquiries from the professors / lecturers on which venues are available in the campus during certain timings on a certain day. You are to write a function available\_venues which takes as input a file, a start time, an end time, and a day code and it returns you a list of all the venues that are available during that duration on that particular day. [10 marks]

Finding out which are the rooms available on Day 4, during 1000 – 2200:

>>> available\_venues("timetable2.csv", 1000, 2200, 4)

['CAPT-SR4', 'CAPT-SR6', 'CEProj', 'CMS', 'COM1-0208', 'COM1-0212', 'COM1-0217', 'COM1-B111', 'COM1-B113', 'DV1', 'E1-06-01', 'E1-06-11', 'CELC-SR2A', 'COM1-0203', 'COM2-0108', 'CR4-1', 'DV2', 'E-LAB', 'E1-06-07', 'CAPT-DV', 'COFM-LAB', 'CR1', 'CAPT-SR2', 'COM1-B102', 'COM1-B112', 'CR3-4', 'CR4-2', 'DV', 'E1-06-05', 'E2-03-03', 'CELC-SR1A']

>>> len(available\_venues("timetable2.csv", 1000, 2200, 4))

31

**C.** Your boss would like to understand the utilization of the available space in the school based on the timetables. You are to write a function venue\_occupancy that takes as input a file and returns the average venue occupancy for all the venues in the university. For venue occupancy, we consider only office hours, i.e. between 800 and 1700, for weekdays (Mondays to Fridays).

Venue occupancy is the ratio of occupied hours to the total available hours. Hint: You should look at columns 4, 5 and 6 of the provided data file. Since you are working with floating point numbers in this problem, there are likely to be some floating point errors. As long as your answer is to within 3 decimal places of the actual answer, your answer would be considered to be correct. [10 marks]

Also, while optimality is a core goal, your function is expected to be able to run reasonably fast and it should finish the computation within 30 seconds for a “regular” PC.

Sample execution:

>>> venue\_occupancy("timetable1.csv")

0.14354005167958656

>>> venue\_occupancy("timetable2.csv")

0.12550860719874804

Hint: You probably want to check your work as you go along instead of writing all your code and trying to debug everything at once. Consider yourself warned.

**D**. Assuming that the valid booking hours for all venues are between 0600hrs to 2400hrs, write a function, available\_timing which takes as input a filename, a day code, a venue string, and it will return the timings when the venue is available.

>>>available\_timing(“timetable2.csv”, 5, “CR1”)  
[(600, 1200), (1800, 2400)]

>>> available\_timing(“timetable2.csv”, 3, “CR1”)  
[(600,1400),(1800,2400)]

>>>available\_timing(“timetable2.csv”, 1, “CR1”)  
[(600,2400)]

**E.** Your boss wants you to think of a way to present the availability of the venues in a more ‘graphical way’ so that it is easier to spot the vacant timings instead of having to check the csv file manually. Write one / more functions to enable your program to produce a pretty-printed (pp) graphical output of a particular venue’s bookings and vacant status for the week.

You are not limited to the proposed output format. Also, if you want to use other python packages or tools to come out with prettier outputs, please feel free to do it!

p.s. To make it manageable for your implementation, you may assume that the timings are in hourly intervals, and will not contain partial hour data.

>>> pp\_schedule(“timetable2.csv”, “CR1”)

Venue: CR1  
+-----+-----+-----+-----+-----+------+------+--------+--------+--------+--------+--------+--------+------+------+------+------+------+------+------+

| Day | 600 | 700 | 800 | 900 | 1000 | 1100 | 1200 | 1300 | 1400 | 1500 | 1600 | 1700 | 1800 | 1900 | 2000 | 2100 | 2200 | 2300 | 2400 |

+-----+-----+-----+-----+-----+------+------+--------+--------+--------+--------+--------+--------+------+------+------+------+------+------+------+

| Mon | | | | | | | | | | | | | | | | | | | |

| Tue | | | | | | | | | | | | | | | | | | | |

| Wed | | | | | | | | | LA5031 | | | | | | | |

| Thu | | | | | | | | | | | | | | | | | | | |

| Fri | | | | | | | PF2401 | PF2401 | AR5421 | AR5421 | AR2223 | AR2223 | | | | | | | |

| Sat | | | | | | | | | | | | | | | | | | | |

| Sun | | | | | | | | | | | | | | | | | | | |

+-----+-----+-----+-----+-----+------+------+--------+--------+--------+--------+--------+--------+------+------+------+------+------+------+------+