2020 Take-Home Examination for Open Software Practice

D.S. Hwang, Dankook University

Attention

- Your code must run on Python 3.6 or higher.
- Your document should include problem, code and captured screen.
- Your source code should be emailed along with your document.

Problem 1

The file `A.dat` store each value with its row and column indices by the following format.

```
1  row_idx,column_idx:value
```

The data is stored in unordered sequence and line by line. You need to read all the element into a matrix for the input file for the following problems. An appropriate regular expression is recommended to easily read matrix elements.

```
 1  55,0:6 18,4:2 67,2:-9 36,3:1 54,1:1
 2  51,2:-3 58,3:1 26,3:9 62,4:5 22,4:-6
 3  5,0:6 96,3:-5 21,2:-6 8,4:1 61,3:-8
 4  64,3:5 34,1:-6 29,1:-10 44,0:-1 63,1:-3
 5  50,3:10 23,1:9 42,2:-5 79,1:-10 99,2:8
 6  0,1:7 45,1:-8 50,0:5 32,4:8 73,3:10
 7  5,3:0 12,1:-10 94,1:6 59,3:4 13,4:-2
 8  77,3:5 15,3:2 33,2:-9 95,4:-4 5,2:9
 9  63,0:-4 7,4:4 49,3:-3 34,0:10 65,0:-1
10                  :
```

1. (10pt.) Write and test your Python function to find the maximum values of row and column(file name: find_size.py).
2. (10pt.) Write and test your Python function to find the maximum and minimum values(file name: find_value.py).
3. (10pt.) Write and test your Python function to calculate the Euclidean distance between 10-th and 27-th rows(file name: comp_dist.py).
4. (20pt.) Write and test a Python function that finds row indices whose Euclidean distance is less than or equal to 10.0 to the 37th row(file name: find_index.py).

Problem 2

Modularization is a key for software design and implementation. This will evaluate your understandability on the modularization concept. The code `expgm.c` has some functions written in C.

1. (5pt.) Compile and execute the program on a command line.
2. (5pt.) Describe what this program does.
3. (10pt.) Split `expgm.c` into more than 6 files. Each file should include related functions or macros. Write and test your Makefile script to build the executable file.
4. (10pt.) Based on your split sources, write and test your CMake script to build the executable file.

Problem 3

(10pt.) Write a shell script to extract file name and size in the current directory. The output should display the file name first, followed by the file size like this example(file name: script.sh).

```
1 dsscript.sh 64
2 expgm 13032
3 expgm.c 2459
4 total 111811
```

Problem 4

(10pt.) Design and implement a Python script that reads a set of integers and returns the integers that occur two or more times(file name: dup.py). The input is 10, 9, 8, 7,6, 5, 4, 3, -3, -3, 4, 4.

Problem 5

(10pt.) Please list all the contents you have leaned through this class. In addition, point out what the professor should improve to make this class better.