

Computational Management Science 1

Final

registration number: SAMPLE SOLUTION

(Do not write your name on the test - just the 7 digit student id number.)

1. (6 points) Writing Code

(a) (3 points, ≤ 5 minutes) Functions

Write a function `productsum(first_iterable, second_iterable)` in Python that takes a two sequences of numbers with any length. The function must calculate the product of the elements of each of the sequences and return the sum of the two products. [e.g. `productsum((1, 3, 4, 1), (9, 2, 3)) \Rightarrow 12+54=66;` `productsum([3, 5], [2, 4]) \Rightarrow 23`]. Add a proper docstring to receive full points.

```
def productsum(first_iterable, second_iterable):  
    """Return the sum of the inner products of each iterable."""  
    first, second = 1, 1  
    for val in first_iterable:  
        first *= val  
    for val in second_iterable:  
        second *= val  
    return first + second
```

(b) (3 points, ≤ 5 minutes) Classes and Data Structures

Implement a simple data structure in Python. The data structure must be capable of storing a table with named rows and columns. Write a **minimalistic** class. You don't need to implement any functionality, not even the functions to access / index the data. The class must be initialized with row and column labels and all values must be set to 0.0. [e.g. `a_table = Table(['row1', 'row2'], ['col1', 'col2', 'col3'])`]. Don't forget to write docstrings in order to receive full points.

```
class Table(object):  
    """A table with named rows and columns."""  
    def __init__(self, row_labels, col_labels):  
        row = {label: 0.0 for label in col_labels}  
        self.__data = {label: dict(row) for label in row_labels}
```

2. (6 points, ≤10 minutes) Correct Mistakes

The following code contains 6 syntax errors/ typos. Clearly mark and correct the mistakes. (hint: you don't need to understand what the function does to correct the mistakes as there are no logical errors; assume that all required classes are available → just look for syntax errors)

```
class SortationCenter(OptimizationObject):  
    """Sortation center representation for transportation problems."""  
    def __init__(self, sortation_center: dict, fulfillment_centers: dict,  
                customers: dict):  
        self.name = sortation_center['name']  
        self.unit_costs = {fulfillment_centers[fc['from']]: fc['cost']  
                          for fc in sortation_center['unit_cost']}  
        self.shipping_costs = {customers[cust['to']]: cust['cost']  
                               for cust in sortation_center['shipping_cost']}
```

3. (6 points, ≤10 minutes) Data Analysis Tools

- (a) Name (in total) two advantages and/or disadvantages of pandas over common spreadsheet programs (2p).
 - + pandas performs better for larger datasets
 - + pandas can easily be programmed
- (b) Also name (in total) two advantages and / or disadvantages of pandas over other professional data analysis tools (2p).
 - + pandas uses a permissive open source license (can be used free of charge even in commercial environments)
 - + pandas has a huge community which guarantees continued development
- (c) Finally, name one thing you like about this course and one thing that should be improved in the future (be honest!) (2p). UP TO THE STUDENTS...

4. (12 points) Reading and Understanding Code

What do the following code snippets print to stdout? Write exactly what the output of each snippet is if **the snippet is the only content of a Python file**. If the output is an error message, it is enough to write "ERROR". If there is no output, write "None"

(a) Simple calculation

```
print(3 * 2 * 2 ** 3)
```

48

(b) Pandas

```
from pandas import DataFrame
df = DataFrame([[20, 'name1'], [15, 'name2']])
print(df[1][1])
```

'name2'

(c) Numpy

```
import numpy as np
m = np.arange(4).reshape(2,2)
print(m[1,1])
```

3

(d) Math

```
import math
l = [5, 3, 2]
print(math.pi(l[0], l[1]))
```

ERROR

(e) Lists

```
l1 = [5, 3]
l2 = [3, 5]
l1.append(l2)
print(sum(l1))
```

ERROR

(f) Sorting

```
l = [2, 4, 3]
print(sorted(l))
```

[2, 3, 4]

5. (5 points, ≤5 minutes) PuLP

(a) (3 points)

Name three advantages and / or disadvantages of PuLP over proprietary, designated modelling languages.

+ PuLP gives freedom of using full programming language; + PuLP uses a permissive open source license; - PuLP occasionally suffers from performance problems

(b) (2 points)

List at least two tricks to apply when running out of memory (RAM) when using PuLP and a MILP solver directly from within the same Python program.

make sure you're using generator expressions wherever possible (instead of list comprehensions); have PuLP export the model and start the solver in a separate process

6. (6 points), ≤5 minutes) Pandas

Assume pandas is imported into your namespace as pd.

(a) (1.5 points)

Initialize a pandas series with the values (4, 2, 3, 5, 7) which use the labels ['Sue', 'Pat', 'Chris', 'John', 'Stu']. The series must be stored in a variable named s.

```
s = pd.Series((4, 2, 3, 5, 7), index=['Sue', 'Pat', 'Chris', 'John', 'Stu'])
```

(b) (1.5 points)

Generate basic statistic figures over the values in s and print them.

```
print(s.describe()) # print can be skipped in interactive environment
```

(c) (1.5 points)

Read a compressed csv file into a pandas data frame. The file is called 'pv_data.csv.bz2', directly starts with the column names and has ',' to separate the columns. The data frame must be stored in a variable called df.

```
df = pd.read_csv('pv_data.csv.bz2')
```

(d) (1.5 points)

Print the last couple of lines of df to stdout.

```
print(df.tail()) # print can be skipped in interactive environment
```

7. (7 points) Tools

(a) (3 points)

What is git and what can it do for you? Name some of the key features. Give examples of file types that work well with git and examples of file types that are not ideal.

git allows distributed version control for teams small and large; you automatically get distributed backups, high performance since most operations are local; it allows branching and merging at ease and hence permits modern workflows (eg. feature branches); git works particularly well with text files, binaries are not ideal

(b) (3 points)

Name the three types of UI and give a pro and a contra for each approach.

console (+fast, scriptable; -steep learning curve), GUI (+easier to use; -more effort to code), web (+allows mobility, device independent; -harder to get right)

(c) (1 points)

Is math part of the Python standard library? What about numpy?

math: yes; numpy: no