COMP10001

2023 Sem 1

Foundations of Computing

Week 9

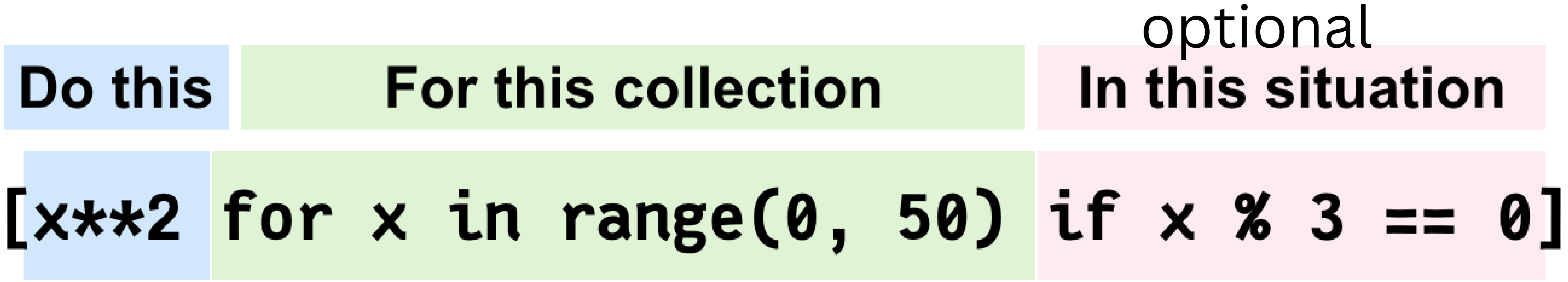
Tutorial/Workshop ○ ○ ○ ○

Today's Tutorial

1. list comprehensions
2. text files
3. csv files

List comprehensions

Shortcut notation to do a simple iteration



optional

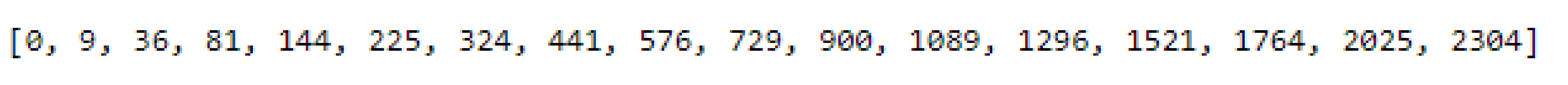
Do this

For this collection

In this situation

```
[x**2 for x in range(0, 50) if x % 3 == 0]
```

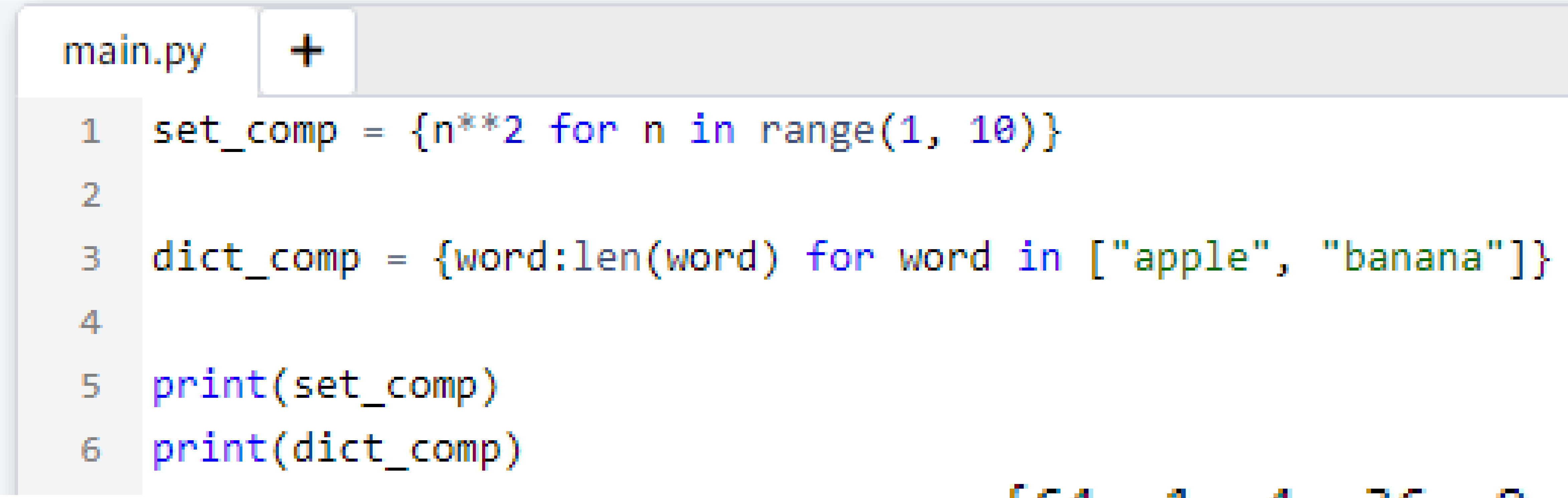
x^2 in the range 0-50 if x is a multiple of 3



```
[0, 9, 36, 81, 144, 225, 324, 441, 576, 729, 900, 1089, 1296, 1521, 1764, 2025, 2304]
```

List comprehensions

Can use them to create sets and dictionaries too

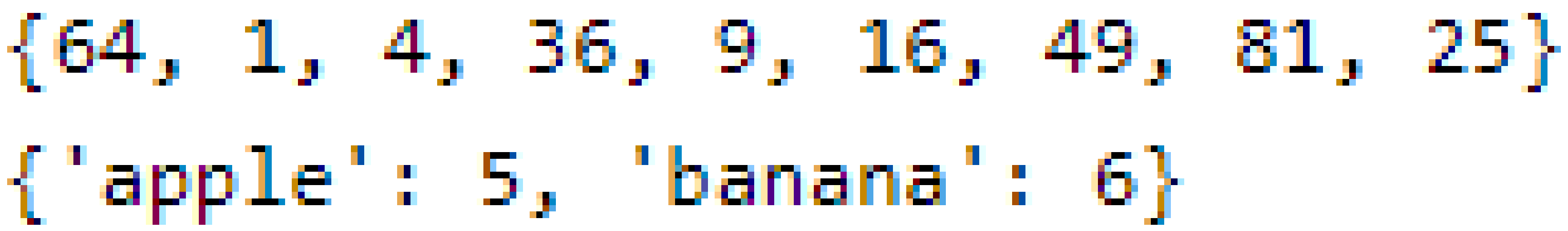


```python
set_comp = {n**2 for n in range(1, 10)}

dict_comp = {word:len(word) for word in ["apple", "banana"]}

print(set_comp)
print(dict_comp)
```

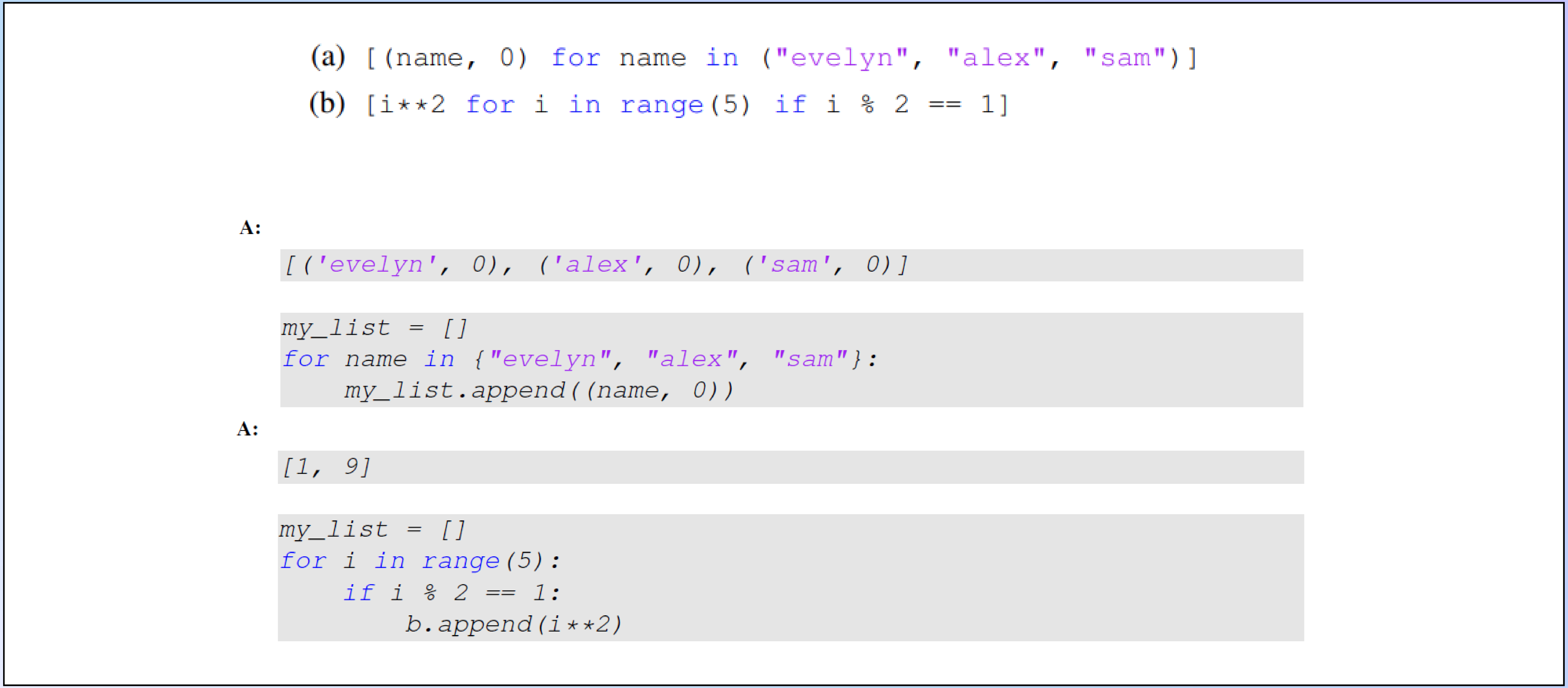
note the lack of order in set



```
{64, 1, 4, 36, 9, 16, 49, 81, 25}
{'apple': 5, 'banana': 6}
```

Exercise 1

Exercise 1 answer



(a) `[(name, 0) for name in ("evelyn", "alex", "sam")]`

(b) `[i**2 for i in range(5) if i % 2 == 1]`

A:

```
[('evelyn', 0), ('alex', 0), ('sam', 0)]
```

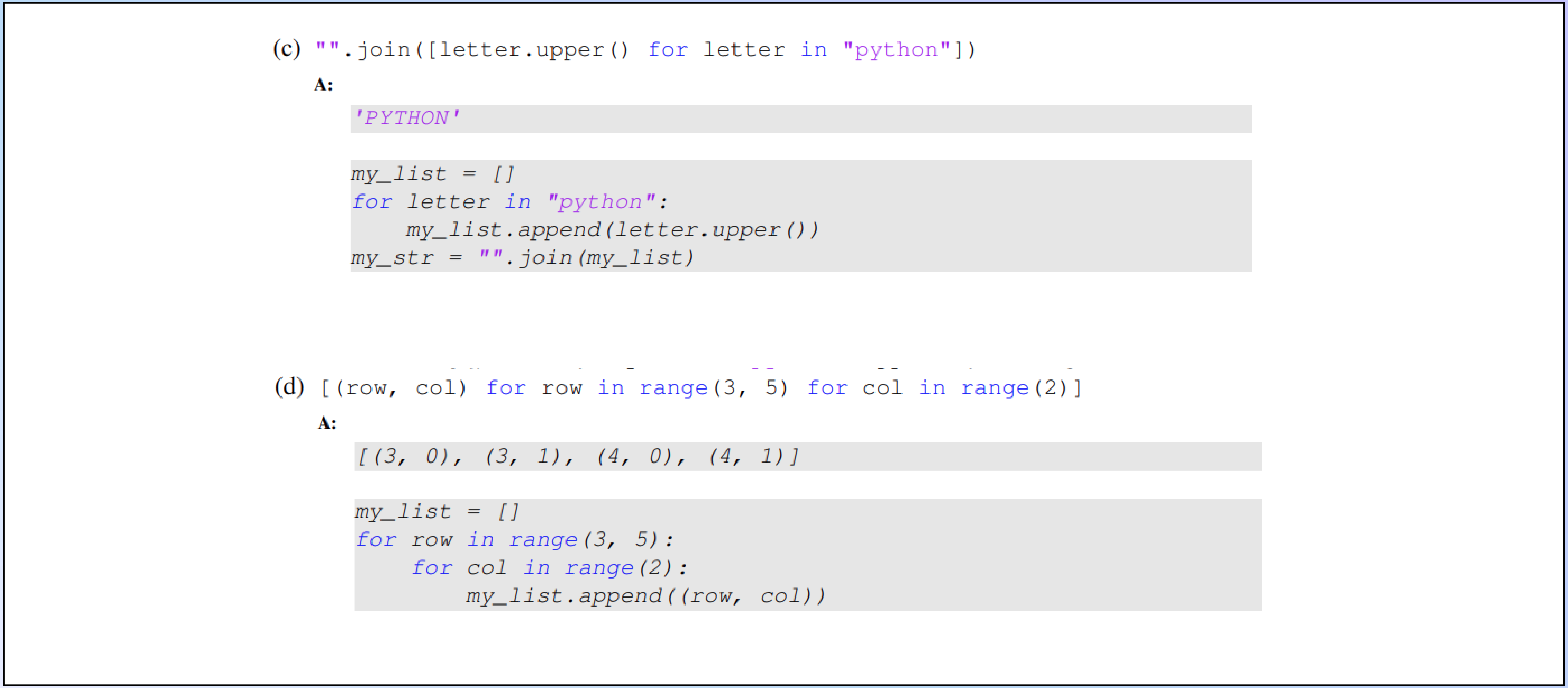
```
my_list = []
for name in {"evelyn", "alex", "sam"}:
    my_list.append((name, 0))
```

A:

```
[1, 9]
```

```
my_list = []
for i in range(5):
    if i % 2 == 1:
        b.append(i**2)
```

Exercise 1 answer



(c) `"".join([letter.upper() for letter in "python"])`

A:

```
'PYTHON'
```

```
my_list = []
for letter in "python":
    my_list.append(letter.upper())
my_str = "".join(my_list)
```

(d) `[(row, col) for row in range(3, 5) for col in range(2)]`

A:

```
[(3, 0), (3, 1), (4, 0), (4, 1)]
```

```
my_list = []
for row in range(3, 5):
    for col in range(2):
        my_list.append((row, col))
```

Files

Allow us to store data on the computer

Unlike lists, dictionaries, sets, which are erased when the program stops running

Opening files

file = open("file.txt", "r") – read

file = open("file.txt", "w") – write (clears the file)

file = open("file.txt", "a") – append (add to file)

Returns a file object; we can now read the file

Writing and reading files

file.read()

returns a string of the whole file

file.readline()

returns a string of 1 line

file.readlines()

returns a list of strings of each line

file.write()

writes a string to the file

file.close()

close the file when we are done

Writing and reading files

file.read()

returns a string of the whole file

"This is some text. This is in the same line.
Now we are on a new line.
And this is the 3rd line."

file.readline()

returns a string of 1 line

"This is some text. This is in the same line."

file.readlines()

returns a list of strings of each line

["This is some text. This is in the same line.", "Now we are on a new line.", "And this is the 3rd line."]

CSV files

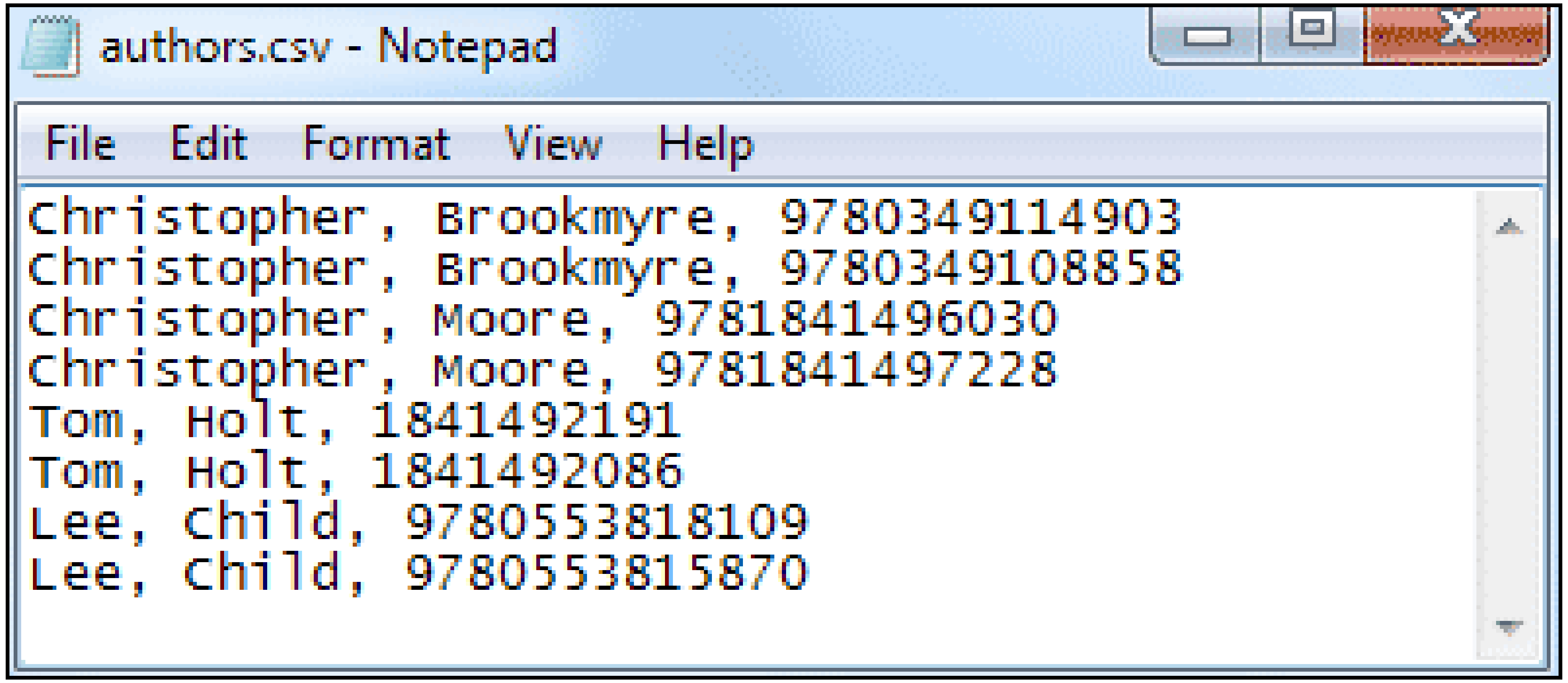
Comma separated files

Similar to spreadsheet

Values separated by ,

Rows separated by \n

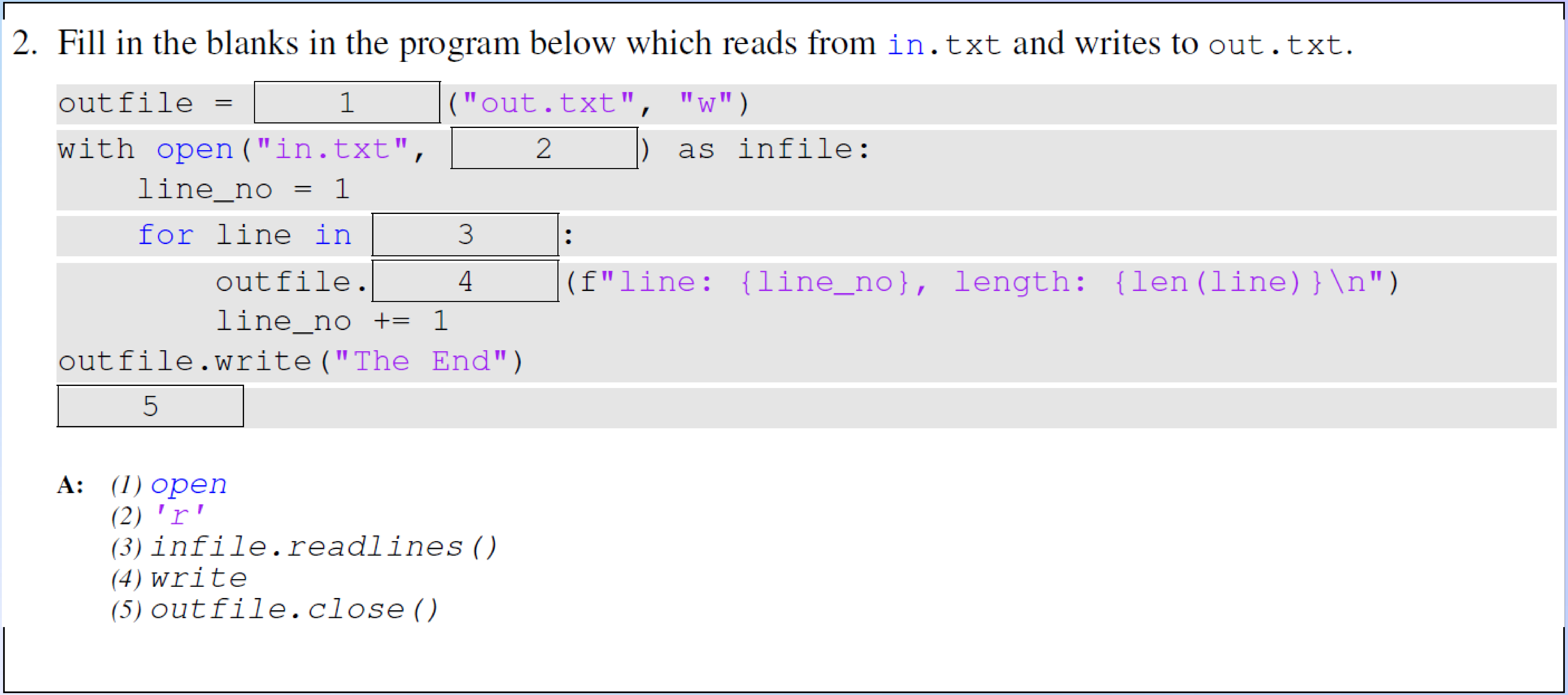
Easy to read and process



```
Christopher, Brookmyre, 9780349114903
Christopher, Brookmyre, 9780349108858
Christopher, Moore, 9781841496030
Christopher, Moore, 9781841497228
Tom, Holt, 1841492191
Tom, Holt, 1841492086
Lee, Child, 9780553818109
Lee, Child, 9780553815870
```

Exercises 2-3

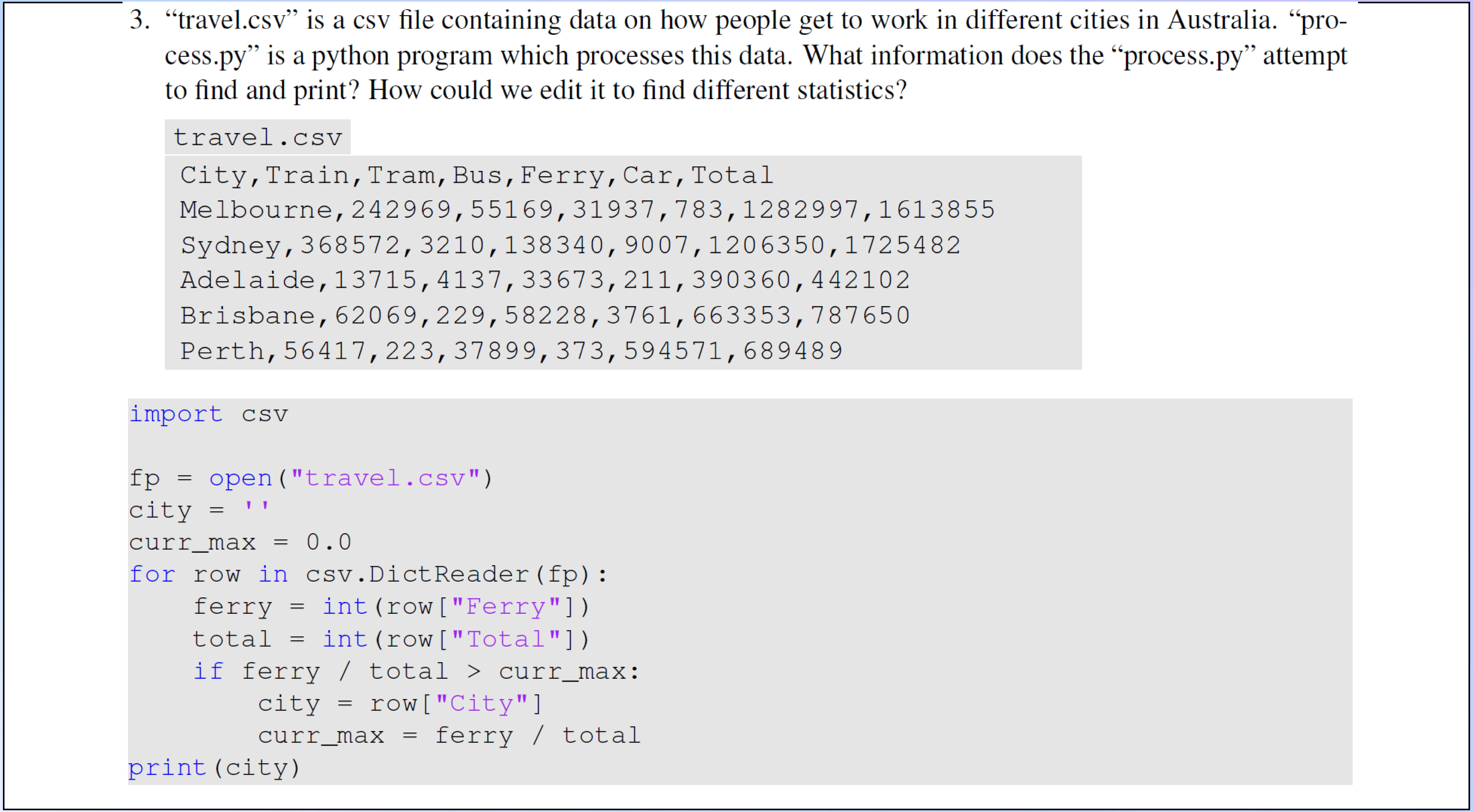
Exercise 2 answer



2. Fill in the blanks in the program below which reads from `in.txt` and writes to `out.txt`.

```
outfile =     1     ("out.txt", "w")
with open("in.txt",     2     ) as infile:
    line_no = 1
    for line in     3     :
        outfile.    4     (f"line: {line_no}, length: {len(line)}\n")
        line_no += 1
outfile.write("The End")
    5
```

A: *(1)* *`open`*
(2) *`'r'`*
(3) *`infile.readlines()`*
(4) *`write`*
(5) *`outfile.close()`*



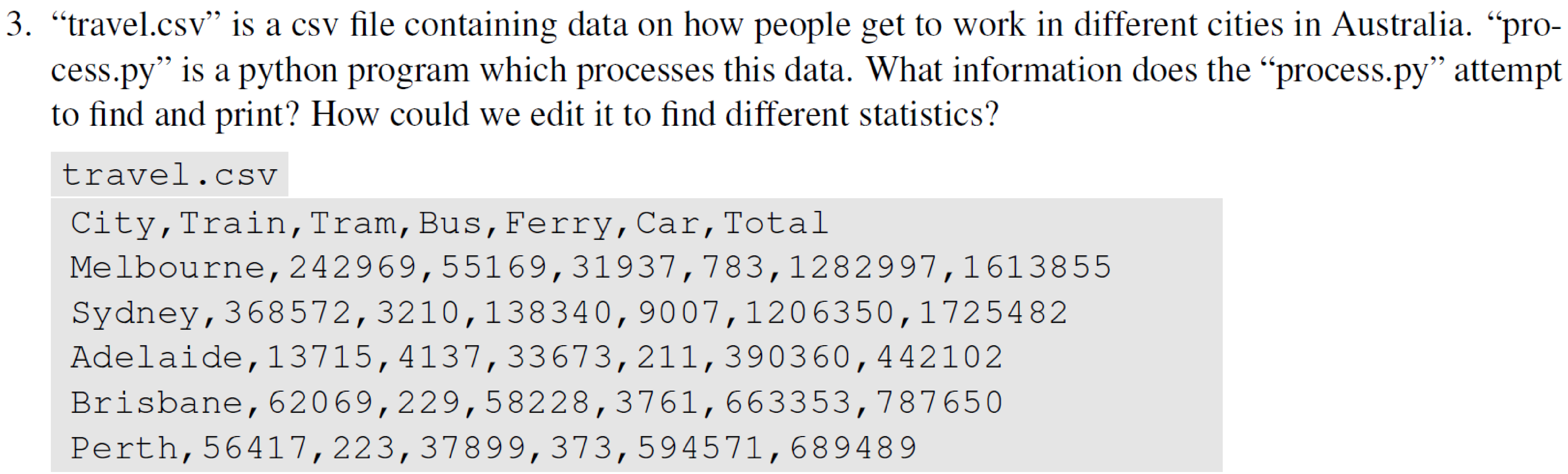
3. “travel.csv” is a csv file containing data on how people get to work in different cities in Australia. “process.py” is a python program which processes this data. What information does the “process.py” attempt to find and print? How could we edit it to find different statistics?

travel.csv

```
City,Train,Tram,Bus,Ferry,Car,Total
Melbourne,242969,55169,31937,783,1282997,1613855
Sydney,368572,3210,138340,9007,1206350,1725482
Adelaide,13715,4137,33673,211,390360,442102
Brisbane,62069,229,58228,3761,663353,787650
Perth,56417,223,37899,373,594571,689489
```

```python
import csv

fp = open("travel.csv")
city = ''
curr_max = 0.0
for row in csv.DictReader(fp):
    ferry = int(row["Ferry"])
    total = int(row["Total"])
    if ferry / total > curr_max:
        city = row["City"]
        curr_max = ferry / total
print(city)
```



3. “travel.csv” is a csv file containing data on how people get to work in different cities in Australia. “process.py” is a python program which processes this data. What information does the “process.py” attempt to find and print? How could we edit it to find different statistics?

travel.csv

```
City,Train,Tram,Bus,Ferry,Car,Total
Melbourne,242969,55169,31937,783,1282997,1613855
Sydney,368572,3210,138340,9007,1206350,1725482
Adelaide,13715,4137,33673,211,390360,442102
Brisbane,62069,229,58228,3761,663353,787650
Perth,56417,223,37899,373,594571,689489
```

- OrderedDict([('City', 'Melbourne'), ('Train', '242969'), ('Tram', '55169'), ('Bus', '31937'), ('Ferry', '783'), ('Car', '1282997'), ('Total', '1613855')])
- OrderedDict([('City', 'Sydney'), ('Train', '368572'), ('Tram', '3210'), ('Bus', '138340'), ('Ferry', '9007'), ('Car', '1206350'), ('Total', '1725482')])
- OrderedDict([('City', 'Adelaide'), ('Train', '13715'), ('Tram', '4137'), ('Bus', '33673'), ('Ferry', '211'), ('Car', '390360'), ('Total', '442102')])
- OrderedDict([('City', 'Brisbane'), ('Train', '62069'), ('Tram', '229'), ('Bus', '58228'), ('Ferry', '3761'), ('Car', '663353'), ('Total', '787650')])
- OrderedDict([('City', 'Perth'), ('Train', '56417'), ('Tram', '223'), ('Bus', '37899'), ('Ferry', '373'), ('Car', '594571'), ('Total', '689489')])

WORKSHOP

○ ○ ○ ○

Grok, problems from sheet, ask me questions :)

See you next week!