Degree in Software Engineering – Computing Basics

Solution to Unit 2.6 Exercises: Files

This document includes the solution to the exercises of the document “Unit 2.6 Exercises: Files”. It is recommended that you try doing the exercises without looking at the solutions first, and then you check your answers. Please note there might be multiple solutions to the same problem.

Exercise 1

Proposed solution:

```
def parse_line(line):
    fields = line.split(",")
    fields[1] = int(fields[1])
    fields[2] = int(fields[2])
    return fields

f = open("input.csv", "r", encoding="utf8")
line = f.readline()
while line != "":
    fields = parse_line(line)
    print(fields)
    line = f.readline()
f.close()
```

Exercise 2

Proposed solution:

```
def parse_line(line):
    fields = line.split(",")
    fields[1] = int(fields[1])
    fields[2] = int(fields[2])
    return fields

result = []
f = open("input.csv", "r", encoding="utf8")
line = f.readline()
while line != "":
    fields = parse_line(line)
    if fields[2] > fields[1]:
        result.append(fields[0])
    line = f.readline()
f.close()

print("There are {} provinces with #women > #men:".format(len(result)))
for province in result:
    print(province)
```

Exercise 3

Proposed solution:

```
def parse_line(line):
    fields = line.split(",")
    fields[1] = int(fields[1])
    fields[2] = int(fields[2])
    return fields

f_in = open("input.csv", "r", encoding="utf8")
f_out = open("output.csv", "w", encoding="utf8")
line = f_in.readline()
while line != "":
    fields = parse_line(line)
    f_out.write("{},{}\n".format(fields[0], fields[1] + fields[2]))
    line = f_in.readline()
f_in.close()
f_out.close()
```