

# NSSA 220

## Task Automation with Interpreted Languages

### Python Exercises

**Instructor: Dr. Fahed Jubair**  
**RIT DUBAI**

# Exercise 1

---

- Write a Python program, called `exercise1.py`, that receives one input argument: a path, and then the program will print “file” if the given path is a file, print “directory” if the given path is a directory, and print “invalid” if no file/directory exists in the given path

```
$ python exercise1.py /usr/lib
```

```
Directory
```

```
$ python exercise1.py /Users/fjubair/Desktop/sampleddata.txt
```

```
File
```

```
$ python exercise1.py /root
```

```
Invalid
```

# Exercise 2

---

- Write a Python program, called exercise2.py, that prompts the user to enter an integer N, and the program will print a triangle of stars. See the below example run.

```
$ python exercise2.py
```

```
Enter an integer: 5
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

# Exercise 3

---

- Write a Python program, called `exercise3.py`, that reads `student_marks.csv` and prints its content on the screen, while replacing all commas with spaces. See the below example run.

```
$ python exercise3.py
```

```
Name Email Mark
```

```
Mark mark@rit.edu 90
```

```
Noora noora@rit.edu 80
```

```
Lara lara@rit.edu 65
```

```
Ahmad ahmad@rit.edu 74
```

```
Terry terry@rit.edu 62
```

```
Fred fred@rit.edu 91
```

```
Susie susie@rit.edu 64
```

# Exercise 4

---

- Write a Python program, called exercise4.py, that generates three random integers between 0 and 99, print them on the screen, and then print the maximum number out of these three numbers. See the below example run.

```
$ python exercise4.py  
random numbers: 25, 74, 85  
max is 85
```

# Exercise 5

---

- Using only list comprehensions, write a program, called `exercise5.py`, that creates three lists: the first list contains all even numbers between 0 and 98, the second list contains all odd numbers between 1 and 99, and the third list contains the summation of the first and second lists. The program then prints all three lists. See the below example run.

```
$ python exercise5.py
```

```
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98]
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]
```

```
[1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65, 69, 73, 77, 81, 85, 89, 93, 97, 101, 105, 109, 113, 117, 121, 125, 129, 133, 137, 141, 145, 149, 153, 157, 161, 165, 169, 173, 177, 181, 185, 189, 193, 197]
```

# Exercise 6

---

- Write a Python program, called `exercise6.py`, that reads `sampledata.txt` and prints only the lines that starts with the letters L, M, and S in alphabetical order. See the below example run.

```
$ python exercise6.py  
Lisa peaches 7  
Mark grapes 39  
Mark watermelons 12  
Susie oranges 12  
Susie oranges 5
```

# Exercise 7

---

- Write a Python program that generates a list of 50 random characters between a-z, and prints them on the screen in a reversed sorted alphabetical order as one string. Name your program exercise7.py. Below is an example run.

```
$ python exercise7.py  
zzyxxxwvvvvvvvuttttsssrqqqqpollkkjjjiiffcccbba
```



# Exercise 8

---

- Write a Python program, called `exercise8.py`, that receives one input argument: a path to a directory, and then the program will print the number of files inside the directory. If the given path is invalid or is a file, then the program must print “Not a directory”. See the below example run.

```
$ python exercise8.py /bin
```

```
35
```

```
$ python exercise8.py /local
```

```
Not a directory
```

# Exercise 9

---

- Write a Python function, called `analyzeString`, that receives one input: a string, and returns two outputs: the number of words and the number of characters in the string, while ignoring white spaces. For example, if the given string is “welcome to python”, then the returned outputs should be 3 and 15.

# Exercise 10

---

- Write a Python function, called `countOfCommonCharacters`, that receives two input strings and returns one output: the number of unique common characters between these two strings. For example, if the given strings are “python” and “kotlin”, then the output should be 3 because the common characters are ‘o’, ‘t’, and ‘n’. Another example, if the given two strings are “noon” and “afternoon”, then the answer should be 2 because the common characters are ‘n’ and ‘o’.
  - An optional challenge: write the function without using *while* or *for* loops.