File Handling in Python

File Operations

File handling consist of four operations:

- Opening File
- Closing File
- Reading from file
- Writing in file

Reading Files

name = open("**filename**")

- opens the given file for reading, and returns a file object

name.read() - file's entire contents as a string

name.readline() - next line from file as a string

name.readlines() - file's contents as a list of lines

- the lines from a file object can also be read using a `for` loop

```
>>> f = open("hours.txt")
>>> f.read()
'123 Susan 12.5 8.1 7.6 3.2\n
456 Brad 4.0 11.6 6.5 2.7 12\n
789 Jenn 8.0 8.0 8.0 8.0 7.5\n'
```

File Input Template

- A template for reading files in Python:

```
name = open("filename")
for line in name:
    statements
```

```
>>> input = open("hours.txt")
>>> for line in input:
...     print(line.strip())   # strip() removes \n

123 Susan 12.5 8.1 7.6 3.2
456 Brad 4.0 11.6 6.5 2.7 12
789 Jenn 8.0 8.0 8.0 8.0 7.5
```

Exercise

- Write a function `input_stats` that accepts a file name as a parameter and that reports the longest line in the file.
  - example input file, `carroll.txt`:

```
Beware the Jabberwock, my son,
the jaws that bite, the claws that catch,
Beware the JubJub bird and shun
the frumious bandersnatch.
```

  - expected output:

```
>>> input_stats("carroll.txt")
longest line = 42 characters
the jaws that bite, the claws that catch,
```

Exercise Solution

```
def input_stats(filename):
    input = open(filename)
    longest = ""
    for line in input:
        if len(line) > len(longest):
            longest = line

    print("Longest line =", len(longest))
    print(longest)
```

Exercise

- Suppose we have this `hours.txt` data:

```
123 Suzy 9.5 8.1 7.6 3.1 3.2
456 Brad 7.0 9.6 6.5 4.9 8.8
789 Jenn 8.0 8.0 8.0 8.0 7.5
```

- Compute each worker's total hours and hours/day.
  - Assume each worker works exactly five days.

```
Suzy ID 123 worked 31.4 hours: 6.3 / day
Brad ID 456 worked 36.8 hours: 7.36 / day
Jenn ID 789 worked 39.5 hours: 7.9 / day
```

Exercise Answer

hours.py

```
1  input = open("hours.txt")
2  for line in input:
3      id, name, mon, tue, wed, thu, fri = line.split()
4
5      # cumulative sum of this employee's hours
6      hours = float(mon) + float(tue) + float(wed) + \
7              float(thu) + float(fri)
8
9      print(name, "ID", id, "worked", \
10           hours, "hours: ", hours/5, "/ day"
```

Writing Files

name = open("**filename**", "w")
name = open("**filename**", "a")

- opens file for write (deletes previous contents), or
- opens file for append (new data goes after previous data)

name.write(**str**) - writes the given string to the file

name.close() - saves file once writing is done

```
>>> out = open("output.txt", "w")
>>> out.write("Hello, world!\n")
>>> out.write("How are you?")
>>> out.close()

>>> open("output.txt").read()
'Hello, world!\nHow are you?'
```

Task

1)Write a Python code, for copying the content of one file to another file.

2)Write a Python code to count the number of words and lines in the input file and display the result.

3)Write a Python program that encrypts the contents of a text file using a simple encryption algorithm. Decrypt the file and display on console.

4)Write a Python program that concatenates the contents of two text files into a single output file.

5)Write a python program to display the length of longest line in the text file and also display the content of that line.

6) Read the student file with data of five students, roll number, name, marks of 5 subjects. Display the contents on console and also calculate the percentage.