ENGR 102 Programming Practice
Practice Session (Week 2)

1. Write a Python program that copies one text file to another text file, omitting any lines that begin with #.

2. Files that hold photographs, videos, zip files, executable programs, etc. are called binary files: they're not organized into lines, and cannot be opened with a normal text editor. Python works just as easily with binary files, but when we read from the file we're going to get bytes back rather than a string. This time, write a Python program that copies one binary file to another. You will need to use rb and wb mode flags where b is for binary.

3. Write a program that searches a directory and all of its subdirectories, recursively, and returns a list of complete paths for all files with a given suffix (like .mp3). Hint: os.path provides several useful functions for manipulating file and path names.

4. Write a function which

    a. Opens the file words.txt

    b. Finds frequency of each letter

    c. Inserts them into the database as (frequency, [letter1, letter2, …, letterN]) pairs

5. Write a function, sumAll(filename), that should read a file containing only integers and return the sum of all these integers.

6. Write a program that reads a file and writes out a new file with the lines in reversed order (i.e., the first line in the old file becomes the last one in the new file.)

7. Write a program that reads a text file and produces an output file which is a copy of the file, except the first five columns of each line contain a four digit line number, followed by a space. Start numbering the first line in the output file at 1. Ensure that every line number is formatted to the same width in the output file. Use one of your Python programs as test data for this exercise: your output should be a printed and numbered listing of the Python program.

8. Write a program that undoes the numbering of the previous exercise: it should read a file with numbered lines and produce another file without line numbers.

9. Create a list of text file names from your current directory to be used for this exercise's purpose. (Hint: you should use the os module function for this, to avoid creating the list by hand.) Then, add a file name to your list manually that does not exist on your directory (to be used to intentionally cause a runtime error). Open each of these files in a loop and be sure to handle the error that the non existing file will cause.

10. Write a program to read a text file and print its output in lowercase letters. In case someone enters another type of file which is not a text file, raise an exception for it. Then, handle the thrown exception, and print the error type and message.

11. The functions provided below are not quite robust, and can crash easily. Work with each function to find out where they can fail. Find out the types of errors, and modify the code to handle them.

```
def exercise1():
    for i in range(1):
        x = int(raw_input("Enter a number: "))
        y = int(raw_input("Enter a second number: "))
        print x,'/','y','=', x/y

def exercise2(a_list):
    sum_of_pairs=[]
    for i in range(len(a_list)):
        sum_of_pairs.append(a_list[i]+a_list[i+1])
    print sum_of_pairs

def exercise3(file_name):
    fileh = open(file_name,'r')
    for line in fileh:
        print line.lower()
    fileh.close()
```

12. Read data from a text file and save each line into a database file, with a random key (*use random module*).

13. Create a text file named "grades.txt" that consists of student names and their scores on the final exam in this format:

    Ahmet 67
    Busra 78
    Omer 67
    Sumeyye 90
    John 67
    Hank 90
    Anna 45

    Then, write a program which opens the file, grades.txt. Next, it stores the name and the corresponding grade of the students in a dictionary {"Ahmet":67,…}. Insert these data into the database, grouping the students which have the same score into one list with the score as the key, e.g., {67:"Ahmet", 90: ["Sumeyye","Hank"]}