BLU537E Homework Assignment # 1

Remarks:

Write the code yourself. ***Cheating is strictly forbidden***.

For each problem write your code in the function format and give the names of the functions as problem numbers, for example for the solution of problem1:

```
def problem1(input):
    return something
```

Put the codes for all problems into one file (jupter notebook file) and name that file using your student username in the following format: badays_blu537e_homework1.ipynb. The notebook file should definitely contain the outputs of the functions, if applicable. Sample solution file (sample_solution.ipynb) is given to you to show how to organize your solutions.

Give as much as documentation for your script using comments.

Note1: For the following problems, **do not** use libraries like Numpy or PANDAS. You are allowed to use only built-in python modules (os, sys, time, collections etc.)

Note2: If your homework solution file has problems in structure you can lose upto ***20*** points!! For example, if you didn't write solutions in function format or if you did not arrange input arguments properly you may lose points.

Problem 1 (20 Points).

This example is taken from a real-life problem. Sometimes, you might need to make small changes to the output of a program. In one situation, I had to change filename of three million files. In this problem, you are going to do the same operation on only small number of files.

You are given a folder named "mol_files" which contain files having ".pdbqt" extension. These files were produced using a scientific software. The file looks like figure given below. In the first line database code of a molecule is given. For example, in the file new1.pdbqt it is "ZINC507664925".

We want to change the file name from new1.pdbqt to "ZINC507664925.pdbqt". Write a function that takes ***the folder name*** as the input and changes the names of the all pdbqt files in that folder. Print the number of files processed. Also, print how many hours would it take to process three million files.

new1.pdbqt — mol_files

```
REMARK  Name = ZINC50764925
REMARK  10 active torsions:
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK    1  A    between atoms: C1_1  and  C2_2
REMARK    2  A    between atoms: C2_2  and  C3_3
REMARK    3  A    between atoms: C3_3  and  C4_4
REMARK    4  A    between atoms: C4_4  and  N1_5
REMARK    5  A    between atoms: C5_6  and  C6_8
REMARK    6  A    between atoms: C11_14  and  C12_16
REMARK    7  A    between atoms: C12_16  and  C13_17
REMARK    8  A    between atoms: C13_17  and  C14_18
REMARK    9  A    between atoms: C14_18  and  C15_19
REMARK   10  A    between atoms: C14_18  and  C16_20
REMARK                            x       y       z     vdW  Elec       q    Type
REMARK                         _______ _______ _______ _____ _____    _____ ____
ROOT
ATOM      1  C   UNL     1       2.343  -4.350   3.846  0.00  0.00    +0.092 C
ATOM      2  C   UNL     1       3.780  -3.824   3.833  0.00  0.00    +0.016 C
ATOM      3  C   UNL     1       4.512  -4.303   5.089  0.00  0.00    +0.020 C
```

The output of your function should be like the following figure.

```
problem1("mol_files")
```

```
the number of processed files: 38
estimated hours to process three million files: 0.169209220953155
```

Problem 2 (20 Points).

A store charges $12 per item if you buy less than 10 items. If you buy between 10 and 99 items, the cost is $10 per item. If you buy 100 or more items, the cost is $7 per item. Write a program that takes how many items are bought as an input and prints the total cost.

```
problem2(15)
```

```
total cost is $150
```

Problem 3 (20 Points).

Write a program that generates a list of 20 random numbers between 1 and 100. When generating random number use random seed as 18.

(a) Print the list.
(b) Print the sorted list (sort list in the descending order)
(c) Print the average of the elements in the list.
(d) Print how many even numbers are in the list.
(e) Print the largest and smallest values in the list.
(f) Print the second largest and second smallest entries in the list

Since we use the same random seed number, your output should be exactly like this:

```
problem3()
```

```
the numbers in the list:
 [24, 16, 85, 58, 43, 31, 26, 63, 81, 64, 24, 62, 38, 59, 34, 26, 33, 89, 16, 42]
the sorted list:
 [89, 85, 81, 64, 63, 62, 59, 58, 43, 42, 38, 34, 33, 31, 26, 26, 24, 24, 16, 16]
the average of the list: 45.7
the number of even-numbers is : 12
the largest element of the list: 89
the smallest element of the list: 16
the second smallest element of the list: 24
the second largest element of the list: 85
```

Problem 4 (20 Points).

The Fibonacci numbers are the sequence below, where the first two numbers are 1, and each number thereafter is the sum of the two preceding numbers. Write a program that takes how many Fibonacci numbers are wanted and then prints that many.

1,1,2,3,5,8,13,21,34,55,89...

```
problem4(5)
```

```
[1, 1, 2, 3, 5]
```

```
problem4(10)
```

```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
```

Problem 5 (20 Points).

Rememer the movie rating example presented in week1. We merged the contents of movies.dat, ratings.dat and users.dat files into one dataframe table using pandas library. Here, in this problem you are asked do the same operation without using pandas library. To get the movie rating data go to this link: https://grouplens.org/datasets/movielens/ and download the ml-1m.zip file.

Write a function that takes three files as input and writes a text file (named as "merged.dat") for merged data. Merged data should have the following order.

user_id movie_id rating timestamp gender age occupation zip title genres Year

Note that we added year information as an extra column!
The input arguments for the functions should be exactly in the following order:

Merged.dat file should be like this:

```
problem5("movies.dat","users.dat","ratings.dat","merged.dat")
```

merged.dat — homework1

```
1::1193::5::978300760::F::1::10::48067::One Flew Over the Cuckoo's Nest ::Drama::1975
1::661::3::978302109::F::1::10::48067::James and the Giant Peach ::Animation|Children's|Musical::1996
1::914::3::978301968::F::1::10::48067::My Fair Lady ::Musical|Romance::1964
1::3408::4::978300275::F::1::10::48067::Erin Brockovich ::Drama::2000
1::2355::5::978824291::F::1::10::48067::Bug's Life, A ::Animation|Children's|Comedy::1998
1::1197::3::978302268::F::1::10::48067::Princess Bride, The ::Action|Adventure|Comedy|Romance::1987
1::1287::5::978302039::F::1::10::48067::Ben-Hur ::Action|Adventure|Drama::1959
1::2804::5::978300719::F::1::10::48067::Christmas Story, A ::Comedy|Drama::1983
1::594::4::978302268::F::1::10::48067::Snow White and the Seven Dwarfs ::Animation|Children's|Musical::1937
1::919::4::978301368::F::1::10::48067::Wizard of Oz, The ::Adventure|Children's|Drama|Musical::1939
1::595::5::978824268::F::1::10::48067::Beauty and the Beast ::Animation|Children's|Musical::1991
1::938::4::978301752::F::1::10::48067::Gigi ::Musical::1958
1::2398::4::978302281::F::1::10::48067::Miracle on 34th Street ::Drama::1947
1::2918::4::978302124::F::1::10::48067::Ferris Bueller's Day Off ::Comedy::1986
```

One final warning:

If a function like problem5 requires a file or folder write your function work appropritately independently from the path provided by the user.

For example, if the required files for problem5 are in a folder named "data-files" in my computer I should be able to run the function for problem5 like this:

```
problem5("data-files/movies.dat","data-files/users.dat","data-files/ratings.dat","merged.dat")
```