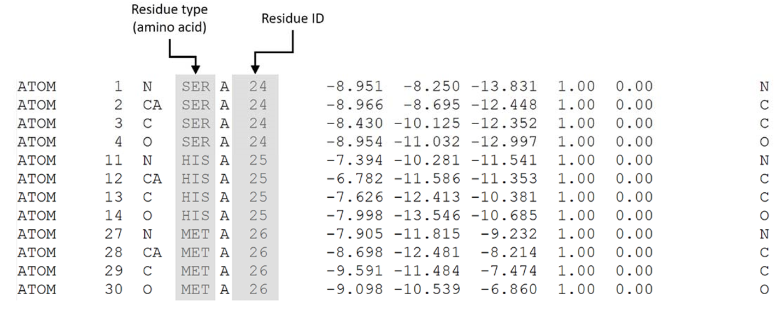
Exercise of Programming Language, Homework E10

Write 4 Python programs to solve the following questions. Please name your program files as *Q1.py*, *Q2.py*, and so on, *i.e.*, according to the serial number of questions. All data files you need can be obtained from the e3 system.

1. Create two target folders, one is DNA and another is Prot. Copy each .seq file to the target folder according to the type of sequence it contains (DNA sequences to the DNA folder; protein sequences to the Prot folder). After copying, the original .seq files should be removed.
2. Protein Data Bank (PDB) is a database for researchers to deposit the protein structures they determined. Structure data of PDB are stored in the PDB file format. In a PDB file, information of an atom is described by a line starting with “ATOM”. As shown in the following figure, in an ATOM line, the “residue type” and “residue ID” of the atom is saved at positions “18–20” and “23–27”, respectively. Because a residue is composed of several atoms, it usually takes several ATOM lines to describe a residue. For instance, in the following figure, there are three residues, serine 24, histidine 25 and methionine 26.



There are 9 PDB files in the PDB folder. Write a program to count the number of residues each file has and print the results to the screen. Your output may look like this (no need to sort the results):

1mp1A.pdb: 111 residues

1or4A.pdb: 169 residues

… (etc.)

1. Use **recursion** to write a function that take two arguments, one is the target folder and another is its output file. What this function do is reading all files and their contents in target folder, taking out all the contents, and then rewriting them into an output file. You may use the Sequence folder to test your function which should successfully make an output named all.fasta.

(Noting that under the target folder may have several subdirectories, each subdirectory may also have its subdirectory as well. All the file contents should be taking out either in target file or in its subdirectory.)

1. Please use **self-defined** **function** and **dictionary** to write a program that computes how many times each “word” occurs in the input sequence. For example:

If two witches would watch two watches, which witch would watch which watch?

Please sort the “words” in your final output (separated with TAB), which should look like this:

Word Frequency

If 1

Two 2

watch 3

watches 1

which 2

witch 1

witches 1

would 2

The program should also pass the following article.

I see trees of green, red roses too.

I see them bloom for me and you.

And I think to myself what a wonderful world.

I see skies of blue and clouds of white.

The bright blessed day, the dark sacred night.

And I think to myself what a wonderful world.

The colors of the rainbow so pretty in the sky.

Are also on the faces of people going by.

I see friends shaking hands saying how do you do.

They're really saying I love you.

I hear babies crying, I watch them grow.

They'll learn much more than I'll never know.

And I think to myself what a wonderful world.

Yes I think to myself what a wonderful world.

1. The Tower of Hanoi (also called the Tower of Brahma or Lucas' Tower and sometimes pluralized) is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:
   1. Only one disk can be moved at a time.
   2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
   3. No disk may be placed on top of a smaller disk.

Write a program to solve the Tower of Hanoi and demonstrate the process of disks moving and count the number of moving.

Output example:

How many disks on the table? 3

move disk 1 from A to C

move disk 2 from A to B

move disk 1 from C to B

move disk 3 from A to C

move disk 1 from B to A

move disk 2 from B to C

move disk 1 from A to C

We have to move 7 times.

Hint: this figure shows the process of 3-disk Tower of Hanoi

