Assignment 5: bash scripting exercise 3: `atomcoordinates`

This course is a gateway into the bioinformatics concentration program at Hunter College in the Biology department or Computer Science Department. Eventually, you will be asked to write programs based on the context of information given. This exercise continues your practicing of bash scripting, and continues your journey into *thinking as a programmer*.

A PDB file contains information obtained experimentally about a macromolecule, usually by either *X-ray crystallography, NMR spectroscopy,* or *cryo-electron microscopy.* The PDB file completely characterizes the molecule, providing the three-dimensional positions of every single atom in the file, where the bonds are, which amino acids it contains if it is a protein or which nucleotides if it is a poly-nucleotide such as DNA or RNA, and much more. The information is not necessarily exact; associated with some of this information are confidence values or other measures that indicate the degree to which the researchers believe that the information is accurate.

Sometimes the researchers who created the file were not sure which of several measured positions of atoms to use, and rather than making a decision, they supplied several different models of the molecule's structure. PDB files with multiple models are easy to spot because they have lines that begin with the word `MODEL`.

A protein can be made up of multiple *chains.* A *chain* is a linear sequence of amino acid residues. The same amino acid residue can occur multiple times within a single chain. Therefore in a PDB file, each residue in a chain is given a sequence number that specifies its position in the chain, starting with `1` as the first position.

Some PDB files also have multiple records that represent the same atom within a single model because the researchers who created the file were not sure which of a few measured positions of atoms to use, and instead of creating separate models, they put different choices of position for these atoms. When an atom has more than one position, a specific character in the ATOM record in the file identifies it.

There are two kinds of atom records in a PDB file: `ATOM` records and `HETATM` records.

- `ATOM` records describe the atoms in the molecule itself.
- `HETATM` records are used to describe atoms that are not part of the biological polymer, such as those in the surrounding solvent or in attached molecules.

An `ATOM` record contains several fields, specified by column numbers on the line. The information is located on the line according to the following PDB file specification.

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 – 6	Record name	"ATOM "	
7 – 11	Integer	serial	Atom serial number
13 – 16	Atom	name	Atom name
17	Character	AltLoc	Alternate location indicator
18 – 20	Residue name	resName	Residue name
22	Character	chainID	Chain identifier
23 – 26	Integer	resSeq	Residue sequence number
31 – 38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms
39 – 46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms
47 – 54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms
77 – 78	LString(2)	element	Element symbol, right-justified

The table specifies which columns the data is in, what data type it is, and what it represents. For example, the serial number is up to five digits long and is in columns 7, 8, 9, 10, and 11. The atomic coordinates are fixed decimal real numbers with 3 digits of decimal precision, with x, then y, then z in that order on the line. Atom names can be up to four characters long. For example, carbon atoms that are part of a ring are named CA, CB, CG, and so on, for C-*alpha*, C-*beta*, and C-*gamma* respectively.

Instructions

Write a script called `atomcoordinates` that will accept the pathname of a PDB file as its only command line argument.

Error checking:
The script should check that it has one single command line argument, and that it is a file that the script can read. If either of these conditions is not true, the program should output to the user

- the appropriate user error,
- a how-to-use-me message,

and then exit.
The script is not required to check that the file is in the proper form for the PDB file format.

Given this PDB file, the program must find all lines that start with the word ATOM and will display, for each line that it finds, a line of output containing the atom's *serial number* and *coordinates.* For example, a line in the PDB file that looks like this:

```
ATOM 18 CB GLN A 3 83.556 52.126 45.080 1.00 26.06 C
```

would result in the following output line being displayed:

```
18 83.556 52.126 45.080
```

because the atom's serial number is `18` and its coordinates are `83.556`, `52.126`, and `45.080`. How do you know where this information is? In the PDB file, the data is in specific columns. In particular, the atom's serial number is always in columns `7` through `11`, and the three coordinates start in column `31` and end in column `54`.

Therefore, your script has to extract the serial number and the coordinates from these columns and display them. Your job is to decide which filters can achieve this. This will take some research. Figure out which filters will work the best.

Grading Rubric

This assignment is graded on a 100 point scale.

The script will be graded on its correctness foremost. This means that it does exactly what the assignment states it must do, in detail. Correctness is worth 70% of the grade. Then it is graded on its clarity, simplicity, and efficiency, worth 30% of the grade.

The objectives when writing any script, are

- *clarity* – it should be easy to understand by someone with a basic knowledge of UNIX
- *efficiency* – it should use the least resources possible
- *simplicity* – it should be as simple as possible

Submitting Requirements

- **Due date:** Sunday October 18, 11:59PM Eastern Standard Time
- Late submissions will be docked total points at the rate of 1 point for every day it is late. No submissions will be accepted after Sunday October 25, 11:59PM Eastern Standard Time.
- Accepted formats:
    - Screenshot image(s) in JPEG, PNG, or TIF of the script, its execution for a sampling various input and the resulting output of each execution.
    - A zip archive file containing the `atomcoordinates` script.
      The steps to create a zip archive and to secure-copy files is outlined in the tutorials called '`zip` command for beginners' and '`scp` command for beginners', that are located in the Course Materials section on the course Blackboard.