Python Mini-Project

The Protein Data Bank (PDB) is a well-known resource in bioinformatics that provides information of 3D structures of large biological molecules. The information of this database can be access via its web site (http://www.pdb.org/), using web services or through the FTP protocol (ftp://snapshots.wwpdb.org/20130101/pub/pdb/data/). Besides the method used to obtain the data, PDB provides different formats for it: PDBML, mmCIF, Chemical Components dictionary and PDB.

The PDB file format is the oldest of the distribution formats and therefore the one that more tools have been developed for. The full documentation of this format can be read from: http://www.wwpdb.org/documentation/format33/v3.3.html

A PDB file describes a biological molecule 3D structure and the information there includes: the atomic level(i.e. Where each atom is located in a 3D coordinate system), the primary and secondary structure, and annotations on the structure such as the mapping between the structure and the protein, because most of the time a PDB structure correspond to a part of the protein and not to the full sequence.

The goal of this project is to be able to extract the information from a PDB file and execute basic analysis on it and it will consider PDB files that refer to 3D structures of proteins (The format can also be used for structures of RNA and other macromolecules).
Although there are libraries such as biopython that can process this file format, the objective here is to test your skills for applying the basis of programming in a real bioinformatics problem, therefore the use of modules that are not built-in into python is not allowed.

PDB Format

The general characteristics of a PDB format are:

- Each line represents a record.
- Each character is considered a column. A white space is consider a character.
- Each line is 80 columns wide and is terminated by an end-of-line indicator(\n).
- The first six columns of every line contain a "record name". This must be an exact match to one on the list of records in the documentation: http://www.wwpdb.org/documentation/format33/v3.3.html
- The information on each record varies, you can check the documentation for details.

For more information please read:
http://www.wwpdb.org/documentation/format33/sect1.html
http://en.wikipedia.org/wiki/Protein_Data_Bank_(file_format)

For this project we will focus in the records below. All the examples are based in the structure with PDB Id 3AYU http://www.pdb.org/pdb/download/downloadFile.do?fileFormat=pdb&compression=NO&structureId=2LP1

- HEADER: First line of the entry, contains PDB ID code, classification, and date of deposition.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
HEADER    HYDROLASE/HYDROLASE INHIBITOR           17-MAY-11   3AYU
```

- TITLE: Description of the experiment represented in the entry.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
TITLE     CRYSTAL STRUCTURE OF MMP-2 ACTIVE SITE MUTANT IN COMPLEX WITH APP-
TITLE    2 DRIVED DECAPEPTIDE INHIBITOR
```

- DBREF: Reference to the entry in the sequence database(s).

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
DBREF  3AYU A    1   110  UNP    P08253   MMP2_HUMAN     110    219
DBREF  3AYU A  111   167  UNP    P08253   MMP2_HUMAN     394    450
DBREF  3AYU B    1    10  UNP    P05067   A4_HUMAN       586    595
```

- SEQRES: Primary sequence of backbone residues.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
SEQRES   1 A  167  TYR ASN PHE PHE PRO ARG LYS PRO LYS TRP ASP LYS ASN
SEQRES   2 A  167  GLN ILE THR TYR ARG ILE ILE GLY TYR THR PRO ASP LEU
SEQRES   3 A  167  ASP PRO GLU THR VAL ASP ASP ALA PHE ALA ARG ALA PHE
SEQRES   4 A  167  GLN VAL TRP SER ASP VAL THR PRO LEU ARG PHE SER ARG
SEQRES   5 A  167  ILE HIS ASP GLY GLU ALA ASP ILE MET ILE ASN PHE GLY
SEQRES   6 A  167  ARG TRP GLU HIS GLY ASP GLY TYR PRO PHE ASP GLY LYS
SEQRES   7 A  167  ASP GLY LEU LEU ALA HIS ALA PHE ALA PRO GLY THR GLY
SEQRES   8 A  167  VAL GLY GLY ASP SER HIS PHE ASP ASP ASP GLU LEU TRP
SEQRES   9 A  167  THR LEU GLY LYS GLY VAL GLY TYR SER LEU PHE LEU VAL
SEQRES  10 A  167  ALA ALA HIS ALA PHE GLY HIS ALA MET GLY LEU GLU HIS
SEQRES  11 A  167  SER GLN ASP PRO GLY ALA LEU MET ALA PRO ILE TYR THR
SEQRES  12 A  167  TYR THR LYS ASN PHE ARG LEU SER GLN ASP ASP ILE LYS
SEQRES  13 A  167  GLY ILE GLN GLU LEU TYR GLY ALA SER PRO ASP
SEQRES   1 B   10  ILE SER TYR GLY ASN ASP ALA LEU MET PRO
```

- HELIX: Identification of helical substructures.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
HELIX    1   1 ASP A   27  ASP A   44  1                                  18
HELIX    2   2 LEU A  114  MET A  126  1                                  13
HELIX    3   3 SER A  151  GLY A  163  1                                  13
```

- SHEET: Identification of sheet substructures.

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
SHEET    1   A 2 ASN A   2  PHE A   3  0
SHEET    2   A 2 LEU A 128  GLU A 129 -1  O  GLU A 129   N  ASN A   2
SHEET    1   B 6 ARG A  49  ARG A  52  0
SHEET    2   B 6 GLN A  14  ILE A  19  1  N  ILE A  15   O  ARG A  49
SHEET    3   B 6 ILE A  60  GLY A  65  1  O  ILE A  62   N  ARG A  18
SHEET    4   B 6 SER A  96  ASP A  99  1  O  PHE A  98   N  GLY A  65
SHEET    5   B 6 ALA A  83  PHE A  86 -1  N  HIS A  84   O  HIS A  97
SHEET    6   B 6 ALA B   7  LEU B   8  1  O  LEU B   8   N  ALA A  85
SHEET    1   C 2 TRP A 104  THR A 105  0
SHEET    2   C 2 TYR A 112  SER A 113  1  O  TYR A 112   N  THR A 105
```

Software Description

Your program should display a menu like the following and wait for the input of the user:

```
********************************************************************************
* PDB FILE ANALYZER                                                            *
********************************************************************************
* Select an option from below:                                                 *
*                                                                              *
*      1) Open a PDB File                       (O)                            *
*      2) Information                           (I)                            *
*      3) Show histogram of amino acids         (H)                            *
*      4) Display Secondary Structure           (S)                            *
*      5) Export PDB File                       (X)                            *
*      6) Exit                                  (Q)                            *
*                                                                              *
*                                                            Current PDB: None *
********************************************************************************
:
```

As you see the user has several options, which can be selected by entering the right number or letter (case unsensitive). The options of the program are:

1. *Open a PDB File*: This function allows the user to indicate a PATH for the file to be analyzed. The software should be able to manage errors with the given path (e.g. File not found). The data from the file should be then load in memory, and display error messages in case the file does not follow the PDB Format.
   Execution Example:

```
: O
Enter a Valid PATH for a PDB File: 3AYU.pdb
The File 3AYU.pdb has been successfully loaded
```

   When a file is successfully loaded the bottom right message of the main menu has to display the file name.
   If a file has been loaded before, a message asking for a confirmation of replacing the current file should be displayed.

2. *Information*: A summary of the file information should be displayed. It should include the filename and title. Notice that a PDB file can include information of more than one chain in the same structure.
   When displaying a sequence each line should have a maximum of 50 amino acids. Execution Example:

```
: I
PDB File: 3AYU.pdb
```

```
Title: CRYSTAL STRUCTURE OF MMP-2 ACTIVE SITE MUTANT IN COMPLEX WITH APP-DRIVED
DECAPEPTIDE INHIBITOR
CHAINS: A and B
 - Chain A
    Number of amino acids:   167
    Number of helix:           3
    Number of sheet:           9
    Sequence:  YNFFPRKPKWDKNQITYRIIGYTPDLDPETVDDAFARAFQVWSDVTPLRF
               SRIHDGEADIMINFGRWEHGDGYPFDGKDGLLAHAFAPGTGVGGDSHFDD
               DELWTLGKGVGYSLFLVAAHAFGHAMGLEHSQDPGALMAPIYTYTKNFRL
               SQDDIKGIQELYGASPD
 - Chain B
    Number of amino acids:    10
    Number of helix:           0
    Number of sheet:           1
    Sequence:  ISYGNDALMP
```

3. *Show histogram of amino acids*: This option allows to display a histogram based on the number of times an amino acid is in the sequence. For this option consider all the chains in the file as a single set. The user can choose to order the histogram by different methods. Example:

```
: H
Choose an option to order by:
   number of amino acids - ascending  (an)
   number of amino acids - descending (dn)
   alphabetically - ascending         (aa)
   alphabetically - descending        (da)
order by: aa

Ala ( 15) : ***************
Arg (  7) : *******
Asn (  5) : *****
Asp ( 20) : ********************
Gln (  5) : *****
Glu (  6) : ******
Gly ( 20) : ********************
His (  7) : *******
Ile ( 10) : **********
Leu ( 13) : *************
Lys (  7) : *******
Met (  4) : ****
Phe ( 12) : ************
Pro ( 11) : ***********
Ser (  8) : ********
Thr (  8) : ********
Trp (  4) : ****
Tyr (  9) : *********
Val (  6) : ******
```

4. *Display Secondary Structure*: For each chain in the loaded pdb, print a representation of the secondary structure using the character ‘/’ to represent an amino acid that is part of a helix, ‘|’ for one that is part of a sheet, and ‘-’ for any other. Each line should have a maximum of 80 characters. Over the representation, the sequence should be displayed, and under it, a tag indicating the identifier of the substructure should be aligned. Execution Example:

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
:S
Secondary Structure of the PDB id 3AYU:
Chain A:
(1)
YNFFPRKPKWDKNQITYRIIGYTPDLDPETVDDAFARAFQVWSDVTPLRFSRIHDGEADIMINFGRWEHGDGYPFDGKDG
-||----------||||||-------//////////////////----||||-------||||||---------------
 1A          2B           1                     1B         3B

LLAHAFAPGTGVGGDSHFDDDELWTLGKGVGYSLFLVAAHAFGHAMGLEHSQDPGALMAPIYTYTKNFRLSQDDIKGIQE
--||||---------||||----||------||/////////////-||---------------------//////////
  5B           4B      1C      2C2             2A                     3

LYGASPD
///----

(167)

Chain B:
(1)
ISYGNDALMP
------||--
      6B
(10)
```

5. *Exit*: The user should be asked to confirm to exit in case he wants to save any changes.

```
: Q
Do you want to exit(E) or do you want go back to the menu (M):E
```

Please notice that options from 2 to 6 do not make sense if there is not a file loaded. Define a strategy to deal with this situation. Also notice that the menu should be displayed every time an option finishes its task.

NOTE: Throughout this paper, where example output is given, your solutions' outputs should match the example out exactly, including white spaces, newlines, punctuation, and case. You may assume all white spaces in the paper are space characters and not tabs.