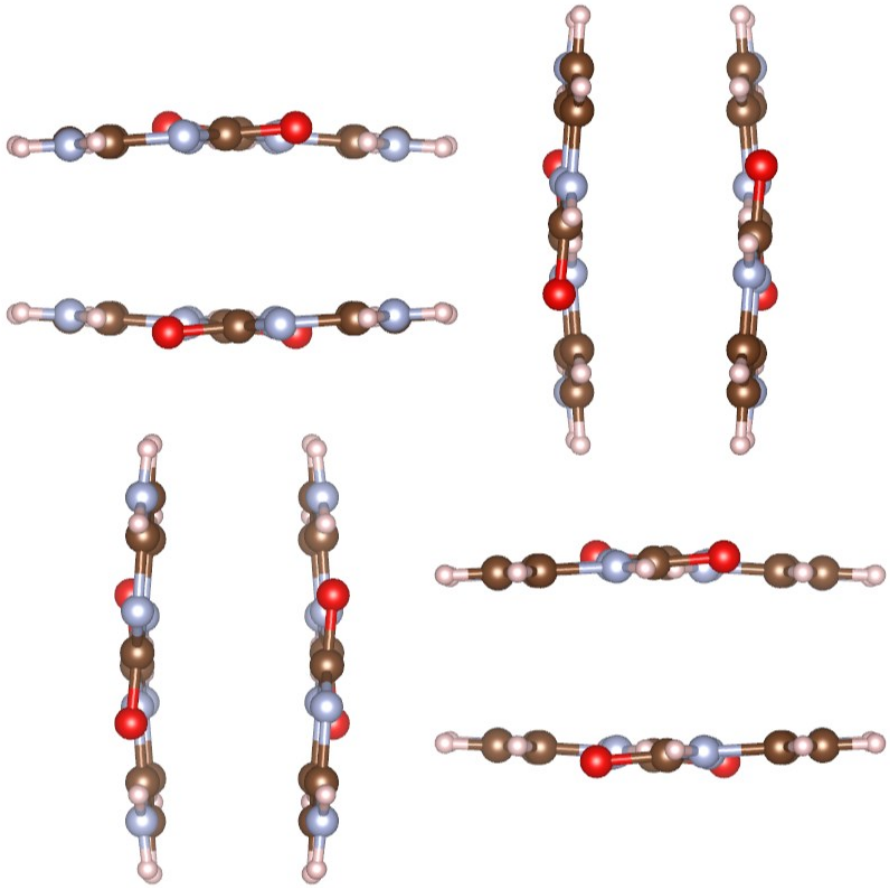
HBM803E - Object Oriented Programming Techniques

FINAL PROJECT (due to Final Exam date)



The aim of this project is to manipulate molecules in the space (rotation, translation).

You will need two files, 16fuCyt.xyz and cell_file. These files can be found in Ninova. You can find information about xyz file format at: https://www.wikiwand.com/en/XYZ_file_format. A simple example:

```
 <number of atoms>
comment line
<element> <X> <Y> <Z>
<element> <X> <Y> <Z>
<element> <X> <Y> <Z>
```

First line of the xyz file is number of atoms in the file. (In the above example, it is 3). Second line is comment line. It could be empty or an information about system. After the second line, every line will consist an element name and its x,y,z coordinates.

Second file, cell_file, is information about a box surrounding the molecule. You can find information about cell box at: https://www.wikiwand.com/en/Primitive_cell In this file, every line is a 3d vector, a_1, a_2, a_3.

In this project, you are expected to handle following steps

- Reading 16fuCyt.xyz file and cell_file,
- Translating and rotating the molecules,
- Finding the volume of the surrounding box,
- Finding the total atomic mass,
- Finding the density.

- The code should be in following form:
    - Cell class
        - Attributes: **a1** -> an array that holds the first line values of the cell_file. **a2** and **a3** are for the other two lines. **volume** holds the volume of the box.
        - Methods:
            - Cell() – constructor that assigns 0 for every attribute
            - Cell(string fileName) – constructor that read values from the given filename(cell_file in this case) and assigns that values to a1,a2,a3 arrays.
            - printCell() – a method the print a1,a2,a3 arrays to the screen. Here you can use printf or cout.
            - calculateVolume() – a method to calculate volume of the box from a1,a2 and a3 arrays. Simply: $V = |a_1 \cdot (a_2 \times a_3)|$
            - getVolume() – a method that return the value of box volume
    - Atom class
        - Attributes: e -> a string that holds element name, x -> holds x coordinate, y-> y coordinate, z-> z coordinate
        - Methods:
            - Atom() – constructor that assigns 0 value to x,y,z and "_" to e
            - Atom(string e, double x, double y, double z) – constructor to assign given values
            - printAtom() – a method to print atom attributes to the screen. You may use cout or printf
            - rotX(double radVal) – a method to rotate atom around x axis with given radian value
            - rotY(double radVal) – a method to rotate atom around

y axis with given radian value
- rotZ(double radVal) – a method to rotate atom around z axis with given radian value

You can find information about rotation at:
http://kwon3d.com/theory/transform/rot.html

- Molecule class
  - Attributes: nAtom -> holds the value of the total atom number, atoms -> a dynamic array that holds atom objects, comment -> holds the comment line of the xyz file, center -> an array that holds x,y,z coordinates of the molecule center, cell -> a cell object, volume -> cell volume, totalMass -> holds the total atomic mass, density -> holds the density information (totalMass/volume)
  - Methods:
    - Molecule(string fileName) – a constructor that reads xyz file assigns nAtom, comment attributes. Then for every line it constructs an atom object and add that object to the atoms array.
    - assignCell(string fileName) – constructs a cell object and assigns it to the cell attribute
    - printMol() – A method to print molecule like a xyz file. Example style:

```
<number of atoms>
comment line
<element> <X> <Y> <Z>
<element> <X> <Y> <Z>
<element> <X> <Y> <Z>
```

    - writeMol(string fileName) – a method to write molecule to a xyz file with given filename
    - findCenter() – a method that finds and assigns the molecule center
    - center2origin() – a method that moves every atom. After this movement new center will be origin.
    - rotX(double radVal) – a method to rotate every atom object in the atoms array around x axis with given radian value
    - rotY(double radVal) a method to rotate every atom object in the atoms array around y axis with given

radian value

- rotZ(double radVal) – a method to rotate every atom object in the atoms array around z axis with given radian value
- getVolume() – a method that return the value of box volume
- printCell() – a method the print a1,a2,a3 arrays to the screen.
- calcTotMass() – a method to calculate total atomic mass of the molecule. You can find atomic mass information at: http://periodictable.com/Properties/A/AtomicMass.an.html
- calcDensity() – a method that calculates density
- getDensity() – a method that returns the value of the density

Do not forget to write destructors for every class and for atoms array. You can visualize xyz files with various programs (e.g. jmol, molden, vmd, vesta and many more.) You can test yourprogram with following main function:

```
int main() {
  Molecule M = Molecule("16fuCyt.xyz");
  M.assignCell("cell_file");
  M.calcDensity();
  cout << "volume = " << M.getVolume() << endl;
  cout << "density = " << M.getDensity() << endl;
  M.printCell();
  M.calcTotMass();
  M.printMol();
  M.findCenter();
  M.center2origin();
  M.printMol();
  for (int i = 0; i < 628; i++) {
      M.rotZ(0.01);
      M.writeMol("rotated.xyz");
  }
  return (0);
}
```

When you test your program with given main function, open rotated.xyz file with molden.(It is free and can be found in nearly every linux repository) When you press the movie button you should see a rotating molecule.

Rules

- As with all programs in this course, your code should contain useful comments. In particular, your name, the date, and a brief description of what the program does should appear at the top of your source file.
- All files must be packed and compressed with following naming convention,

final-project-[STUDENT-ID].tar.gz

The created file must be uploaded to Ninova.