3D Data Processing in Structural Biology
Assignment 2
(due April 25)

Write a program that detects an (almost) largest structural alignment between two proteins or two RNA molecules. Input to the program is two structures in the PDB format.

- Consider only *Cα* atoms for proteins and only phosphate atoms for RNA molecules
- Define transformations using triangles formed by 3 consecutive *Cα* atoms for proteins or phosphate atoms for nucleic acids i.e. try only |*pdb*1|x|*pdb*2| transformations.

Name your program `structAlign.cc` and submit this file in ex2.tar

Input to the program:

```
structalign epsilon pdb1 pdb2
```

where `epsilon` is the largest allowed distance between two corresponding atoms in the alignment

Output:
{Size of the largest alignment} {RMSD between the aligned atoms} {3D transformation (3 rotational and 3 translational parameters)}
For example:
130 1.32 0.5 0.11 0.7 10.1 21.30 -10.1

The transformation transforms `pdb2` on `pdb1`. In addition, your program should create a transformed PDB file '`transformed.pdb`' that contains the transformed `pdb2`. This file should contain **all** atoms (not only *Cα* or *P* atoms).

To help you with this task we provide you a program `alignRand.cc` that performs the alignment task by generating random transformations. It superimposes the centers of mass of the two input PDB files and generates random rotations. You can use this program as a basis for your alignment program.
In addition, we included a number of basic classes, that `alignRand.cc` uses for alignment and of course you can use them too. These classes are a subset of GAMB library: `http://bioinfo3d.cs.tau.ac.il/group/GAMB/GAMB.html`

Briefly, this is the relevant functionality:

- `Vector3` – basic 3D vector class for storing and manipulating xyz coordinates

- `RigidTrans3` – basic class for 3D rigid transformations that stores 3D rotation (`Matrix3` class) and translation (`Vector3` class).
- `Triangle` – basic class for defining triangles and transformations.
- `Atom` – stores the information from the PDB ATOM line that includes index, residue type, coordinates, etc.
- `Molecule` – template container for molecules
- `PDB` – enables efficient PDB parsing. Use PDB::Selector to read specific parts of the PDB.
- `GeomHash` – enables efficient hashing. Useful in building correspondence list, when you look for points that are within short distance from a given point
- `Match` – this class implements a correspondence list. The function `calculateBestFit` computes a transformation that minimizes RMSD for the correspondence list that is stored in the class object.

Sample input files are in the input folder.

Note: you don't have to implement a bipartite matching algorithm to find the maximal correspondence in this assignment. Instead, you can rely on the shortest distance heuristics that is implemented by the `Match` class.

Please submit your code in ex2.tar: `structAlign.cc, Makefile` and a file with a table summarizing alignment size and RMSD for each input pair for `epsilon = 1.0` and `epsilon = 3.0`

Good Luck!