

1 Exercise 1

1. FORMUL record, colum 19 (* for water)
2. 77-78
3. x:31-38, y:39-46, z:47-54
4. 18-20
5. 22 (chainID)

2 Exercise 2

```
1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 #include <vector>
5
6 int main(int argc, char* argv[]) {
7
8     std::string line;
9     std::ifstream pdbfile; //(argv[1], std::ios::in);
10    std::ofstream coordfile; //(argv[2], std::ios::out);
11
12    //Sanity check for correct number of arguments
13
14    switch(argc){
15
16        case 2:
17            pdbfile.open(argv[1], std::ios::in);
18            break;
19
20        case 3:
21            pdbfile.open(argv[1], std::ios::in);
22            coordfile.open(argv[2], std::ios::out);
23            break;
24
25        default:
26            std::cout << "Wrong number of arguments.\n\n Usage:
27            ParsePDB inFile [outFile]" << "\n";
28            return 1;
29    }
```

```

29
30 if (pdbfile.is_open()){
31
32     while (getline(pdbfile,line)){
33
34         if (line.substr(0,4) == "ATOM"){
35
36             //x:31-38, y:39-46, z:47-54; one less because arrays in C
37             ++ are 0-based
38             //a single space is added because it may be, that all
39             coordinates are the maximum allowed size.
40             //no trimming is done. could be added later on.
41             switch(argc){
42                 case 2:
43                     std::cout << line.substr(30,8) << ' ' << line.substr
44                     (38,8) << ' ' << line.substr(46,8) << '\n'; //substr(position,
45                     length); just a reminder for myself
46                     break;
47                 case 3:
48                     coordfile << line.substr(30,8) << ' ' << line.substr
49                     (38,8) << ' ' << line.substr(46,8) << '\n';
50                     break;
51             }
52         }
53     }
54     pdbfile.close();
55     coordfile.close();
56 }
57
58 else std::cout << "Unable to open file";
59
60 return 0;
61 }

```

3 Exercise 3

```

1 #include <iostream>
2 #include <fstream>
3 #include <sstream>
4 #include <vector>
5 #include <string>
6

```

```

7 | int main(int argc, char* argv[]) {
8 |
9 |     std::string line;
10 |    std::ifstream pdbfile;
11 |    std::ofstream translatefile;
12 |    std::vector<double> d;
13 |
14 |    if(argc == 3){
15 |
16 |        pdbfile.open(argv[1], std::ios::in);
17 |        translatefile.open(argv[2], std::ios::out);
18 |        d.assign(3,1.000);
19 |
20 |    }else if(argc == 6){
21 |
22 |        pdbfile.open(argv[1], std::ios::in);
23 |        translatefile.open(argv[2], std::ios::out);
24 |
25 |        d.push_back(std::stod(argv[3]));
26 |        d.push_back(std::stod(argv[4]));
27 |        d.push_back(std::stod(argv[5]));
28 |
29 |    }else{
30 |
31 |        std::cout << "Wrong amount of Parameters\n\n Usage:
    HandlingPDB inFile outFile\n";
32 |        return 1;
33 |
34 |    }
35 |
36 |    //Basic idea:
37 |    // Checking ATOM
38 |    // Grabbing Coordinates based on columns
39 |    // Parsing to double, adding offset
40 |    // And the c++ way to parsing double to string with fixed size
    and precision is to set up an outstream correctly
41 |    // Important: If you want a precision of 3, i.e 3 digits after
    the dot, you need an outstream.precision of 4. Otherwise
42 |    // it get rounded to early.
43 |    // Lastly, replacing the new string in the selected line
44 |    if (pdbfile.is_open()){
45 |        while (getline(pdbfile, line)){
46 |
47 |            if (line.substr(0,4) == "ATOM"){
48 |
49 |                double xCoord = stod(line.substr(30,8));
50 |                double yCoord = stod(line.substr(38,8));
51 |                double zCoord = stod(line.substr(46,8));
52 |

```

```

53
54     xCoord += d[0];
55     yCoord += d[1];
56     zCoord += d[2];
57
58     std::ostringstream xCoordStrs;
59     std::ostringstream yCoordStrs;
60     std::ostringstream zCoordStrs;
61
62     xCoordStrs.width(8);
63     xCoordStrs.precision(4);
64     yCoordStrs.width(8);
65     yCoordStrs.precision(4);
66     zCoordStrs.width(8);
67     zCoordStrs.precision(4);
68
69     xCoordStrs << xCoord;
70     yCoordStrs << yCoord;
71     zCoordStrs << zCoord;
72
73     line.replace(30,8,xCoordStrs.str());
74     line.replace(38,8,yCoordStrs.str());
75     line.replace(46,8,zCoordStrs.str());
76 }
77
78     translatefile << line << '\n';
79
80 }
81
82 }else{
83
84     std::cout << "Unable to open file";
85
86 }
87
88 return 0;
89
90 }

```