

Outputs of program:

1. output in terminal:

pdb file contains 983 lines:

602 lines starting with ATOM

72 lines starting with ATOM containing LEU

76 lines starting with ATOM containing CA

number of references: 4

2. Output.txt

#####lines starting with ATOM at the begining#####

ATOM 1 N MET A 1 27.340 24.430 2.614 1.00 9.67 N

ATOM 2 CA MET A 1 26.266 25.413 2.842 1.00 10.38 C

.
.
.

ATOM 601 O GLY A 76 38.933 40.525 35.687 0.25 36.13 O

ATOM 602 OXT GLY A 76 40.862 39.575 36.251 0.25 36.27 O

#####lines starting containing LEU in 4th column#####

ATOM 60 N LEU A 8 30.132 40.069 18.642 1.00 9.84 N

ATOM 61 CA LEU A 8 29.607 41.180 19.467 1.00 14.15 C

.
.
.

ATOM 580 CG LEU A 73 39.502 37.340 26.002 0.45 30.16 C

ATOM 581 CD1 LEU A 73 38.684 36.647 24.923 0.45 29.57 C

#####lines starting with ATOMcontaining CA in 3rd column#####

ATOM 2 CA MET A 1 26.266 25.413 2.842 1.00 10.38 C

ATOM 10 CA GLN A 2 26.850 29.021 3.898 1.00 9.07 C

.
.
.

ATOM 595 CA GLY A 75 41.845 36.550 32.686 0.25 36.07 C

ATOM 599 CA GLY A 76 40.373 39.813 33.944 0.25 36.19 C

#####distance between each pair of CA#####

distance between 1th CA and 2th CA 3.804452

distance between 1th CA and 3th CA 6.576170

.

.

.

distance between 74th CA and 76th CA 7.071464

distance between 75th CA and 76th CA 3.794274

#####Amino acide sequences#####

1th sequence: MET GLN ILE PHE VAL LYS THR LEU THR GLY LYS THR ILE

2th sequence: THR LEU GLU VAL GLU PRO SER ASP THR ILE GLU ASN VAL

3th sequence: LYS ALA LYS ILE GLN ASP LYS GLU GLY ILE PRO PRO ASP

4th sequence: GLN GLN ARG LEU ILE PHE ALA GLY LYS GLN LEU GLU ASP

5th sequence: GLY ARG THR LEU SER ASP TYR ASN ILE GLN LYS GLU SER

6th sequence: THR LEU HIS LEU VAL LEU ARG LEU ARG GLY GLY

#####References#####

REFERENCE 1

AUTH S.VIJAY-KUMAR,C.E.BUGG,K.D.WILKINSON,R.D.VIERSTRA,

AUTH 2 P.M.HATFIELD,W.J.COOK

TITL COMPARISON OF THE THREE-DIMENSIONAL STRUCTURES OF HUMAN,

TITL 2 YEAST, AND OAT UBIQUITIN

REF J.BIOL.CHEM. V. 262 6396 1987

REFN ISSN 0021-9258

REFERENCE 2

AUTH S.VIJAY-KUMAR,C.E.BUGG,K.D.WILKINSON,W.J.COOK

TITL THREE-DIMENSIONAL STRUCTURE OF UBIQUITIN AT 2.8 ANGSTROMS

TITL 2 RESOLUTION

REF PROC.NATL.ACAD.SCI.USA V. 82 3582 1985

REFN ISSN 0027-8424

REFERENCE 3

AUTH W.J.COOK,F.L.SUDDATH,C.E.BUGG,G.GOLDSTEIN

TITL CRYSTALLIZATION AND PRELIMINARY X-RAY INVESTIGATION OF

TITL 2 UBIQUITIN, A NON-HISTONE CHROMOSOMAL PROTEIN

REF J.MOL.BIOL. V. 130 353 1979

REFN ISSN 0022-2836

REFERENCE 4

AUTH D.H.SCHLESINGER,G.GOLDSTEIN

TITL MOLECULAR CONSERVATION OF 74 AMINO ACID SEQUENCE OF

TITL 2 UBIQUITIN BETWEEN CATTLE AND MAN

REF NATURE V. 255 423 1975

REFN ISSN 0028-0836

#####Nature formatted references#####

Python Code:

```
import sys, re
from math import sqrt

# open pdb file
file = open("1UBQ.pdb", "r")
output = open("output.txt", "w")

lines = file.readlines()

print "\npdb file contains %d lines: " % len(lines)

matrixOfData = []
for line in lines:
    matrixOfData.append(line.split())

print "\npdb file contains %d lines: " % len(lines)
#####
print >> output, "lines starting with ATOM at the begining".center(70, "#"), "\n"
ATOM = []
expression1 = r"^ATOM"
for line in lines:
    if re.search(expression1, line):
        print >> output, line
        ATOM.append(line)
print "%d\t lines starting with ATOM" % len(ATOM)
#####
print >> output, "lines starting containing LEU in 4th column".center(70, "#"), "\n"
expression2 = r"LEU"
i = 0
for atomline in ATOM:
    if re.search(expression2, atomline):
#         if atomline[17:20] == expression2:
            print >> output, atomline
            i+=1
print "%d\t lines starting with ATOM containing LEU" % i
#####
print >> output, "lines starting with ATOM\
containing CA in 3rd column".center(70, "#"), "\n"
CA = []
expression3 = r"CA"
i = 0
for atomline in ATOM:
    if re.search(expression3, atomline):
#         if atomline[13:15] == expression3:
            print >> output, atomline
            CA.append(atomline.split())
            i+=1
print "%d\t lines starting with ATOM containing CA" % i
#####
print >> output, "distance between each pair of CA".center(70, "#"), "\n"

for i in range(len(CA)):
```

```

xi = float(CA[i][6])
yi = float(CA[i][7])
zi = float(CA[i][8])
for j in range(i+1,len(CA)):
    xx = xi - float(CA[j][6])
    yy = yi - float(CA[j][7])
    zz = zi - float(CA[j][8])
    rij = float(sqrt(xx**2+yy**2+zz**2))
    print >> output, "distance between %2.dth CA and %2.dth CA\t\
%f".ljust(35) % (i+1,j+1,rij)
#####
print >> output, "Amino acide sequences".center(70,"#"),"\n"

c = 0
expression4 = r"^SEQRES"
for line in lines:
    if re.search(expression4, line):
        temp = line.split()
        c+=1
        print >> output, "%dth sequence: " % c,
        for i in range(4,len(temp)):
            print >> output, temp[i] ,
        print >> output
#####
c = 0
for line in lines:
    if re.search(r"^REMARK", line):
        c += line.count("REFERENCE")
print "number of references: %d\n" % c
#####
print >> output, "Refrences".center(70,"#"),"\n"

ref = []
def Saveref(x):
    ref.append(x)
    print >> output, x

keys = [r"^TITL\s+",r"^AUTH\s+",r"^REF\s+",r"^REFN\s+"]

for line in lines:
    if re.search(r"^REMARK", line):
        if re.search(r"^REFERENCE",line[11:]):
            Saveref(line[11:])
        else:
            for key in keys:
                if re.search(key,line[12:]):
                    Saveref(line[12:])

for i in range(12):
    print ref[i], "\n"

print >> output, "\n", "Nature formatted references".center(70,"#"), "\n"

auth = []
j = 0

```

```

for i in range(len(ref)):
    if re.search(r"^AUTH", ref[i]):
        if not re.match(r"\d", ref[i][5:6]):
            print ref[i][5:6]
            auth.append((ref[i][7:].rstrip("\s+\n\s+")).split(","))
        else:
            auth[j-1] = (str(ref[i-1][7:].rstrip())+str(ref[i][7:].rstrip())).split(",")
            j+=1

```

```

name = []
for i in range(len(auth)):
    for k in range(len(auth[i])):
        temp = auth[i][k].split(".")
        print temp

```

```

print name

```

```

for a in auth:
    print a

```

```

#for t in auth:
#    print t, "\n"

```

```

#            if line.endswith("REFERENCE\s\d\s+"):
#                prine line
#                continue
#            if line.endswith(r"ISSN\s\d{4}-\d{4}\s+"):
#                print line
#print re.match(r"ISSN\s\d{4}-\d{4}", "ISSN 0022-2836")
#####
#### reference manipulation:

```

```

output.close()
file.close()

```