*# include*<stdio.h>

*# include*<string.h>

*# include*<stdlib.h>

*# include*<ctype.h>

*# include*<math.h>

int main(void)

{

FILE \*fpt1; *// File pointer for file to be opened (3snp.pdb)*

FILE \*fpt2; *// File pointer for output file (created after running the program)*

int num\_cols\_in\_file = 80; *// Maximum 80 columns are in the file*

int max\_word\_length = 5; *// Max length of a word*

int atom\_len = 4; *// Max length of atom column*

int atom\_serial\_no\_len = 5; *// Max length of atom-serial-no column*

int atom\_name\_len = 4; *// Max length of atom name*

int residue\_no\_len = 4; *// Max length of residue number*

int coord\_len = 8; *// Max length of coordinates*

int i, temp\_row\_indicator; *// Counter variables*

int store\_atm\_no[20000], store\_res\_no[20000]; *// to store atom number and residue number*

temp\_row\_indicator = 0; *// init to 0 to point to the first row*

char line[num\_cols\_in\_file], word[max\_word\_length];

char test1[6], temp\_array[9], slno[6], atm\_nm[4];

char res\_no[4], store\_res\_nm[20000][4], chain\_id[20000];

char res\_nm[5], store\_atm\_nm[20000][4];

float store\_x[20000], store\_y[20000], store\_z[20000];

float tot\_x, tot\_y, tot\_z, cm\_x, cm\_y, cm\_z; *// Open the file 3snp.pdb in read mode*

fpt1 = fopen("3snp.pdb", "r"); *// Output file named "out.pdb"*

fpt2 = fopen("out.pdb", "w"); *// opening a new output file in write mode*

*while*( fgets(line, num\_cols\_in\_file, fpt1) != NULL )

{

*for* (i = 0; i < (atom\_len); i++)

{

word[i] = line[i]; *// store the word (i.e. "ATOM")*

}

word[atom\_len] = '\0'; *// end the line with '\0'*

*if* ( strcmp(word, "ATOM") == 0 )

{

*for* ( i = 0; i < atom\_serial\_no\_len; i++ )

{

slno[i] = line[i+6];

}

slno[5] = '\0'; *// NULL CHARACTER at the end of the string*

store\_atm\_no[temp\_row\_indicator] = atoi(slno); *// convert string to integer*

*for* ( i = 0; i < 3; i++ )

{

atm\_nm[i] = line[i + 13];

}

atm\_nm[3] = '\0'; *// NULL CHARACTER at the end of the strin*

strcpy( store\_atm\_nm[temp\_row\_indicator], atm\_nm );

*for* ( i = 0; i < 3; i++ )

{

res\_nm[i] = line[i+17];

}

res\_nm[3] = '\0'; *// NULL CHARACTER at the end of the string*

strcpy( store\_res\_nm[temp\_row\_indicator], res\_nm );

chain\_id[temp\_row\_indicator] = line[21]; *// according to the format*

*for* ( i = 0; i < residue\_no\_len; i++ )

{

res\_no[i] = line[i+22];

}

res\_no[residue\_no\_len] = '\0';

store\_res\_no[temp\_row\_indicator] = atoi(res\_no);

*for* ( i = 0; i < coord\_len; i++ )

{

temp\_array[i] = line[i+30];

}

temp\_array[coord\_len] = '\0';

store\_x[temp\_row\_indicator] = atof(temp\_array);

*for* ( i = 0; i < coord\_len; i++ )

{

temp\_array[i] = line[i+38];

}

temp\_array[coord\_len] = '\0';

store\_y[temp\_row\_indicator] = atof(temp\_array);

*for* ( i = 0; i < coord\_len; i++ )

{

temp\_array[i] = line[i+46];

}

temp\_array[coord\_len] = '\0';

store\_z[temp\_row\_indicator] = atof(temp\_array);

temp\_row\_indicator++;

}

}

*for* ( i = 0; i < temp\_row\_indicator; i++ )

{

fprintf(

fpt2,

"ATOM %5d %s %s %c%4d %8.3f%8.3f%8.3f\n",

store\_atm\_no[i],

store\_atm\_nm[i],

store\_res\_nm[i],

chain\_id[i],

store\_res\_no[i],

store\_x[i],

store\_y[i],

store\_z[i]);

}

*// --- Calculate COM --- //*

tot\_x = tot\_y = tot\_z = 0.0;

*for* ( i = 0; i < temp\_row\_indicator; i++ )

{

tot\_x = tot\_x + store\_x[i];

tot\_y = tot\_y + store\_y[i];

tot\_z = tot\_z + store\_z[i];

}

cm\_x = tot\_x / temp\_row\_indicator;

cm\_y = tot\_y / temp\_row\_indicator;

cm\_z = tot\_z / temp\_row\_indicator;

fprintf(

fpt2, "ATOM 99999 ZZZ Z Z 99 %8.3f%8.3f%8.3f\n",

cm\_x,

cm\_y,

cm\_z);

*// Close the files*

fclose(fpt1);

fclose(fpt2);

*return* 0;

}