

付録

プログラムのソースファイル

学籍番号：07s2009n 氏名：小林 伸悟

2011 年 1 月 12 日

1 hita.exeのソースファイル

この節のプログラムの詳細を解説している。アルゴリズムの理解やプログラムの修正などが必要な場合、コメントの書き込みや振られている番号 (A-1)などを参考にしてもらいたい。

head.h

```
1 #ifndef HEAD_INCLUDED
2 #define HEAD_INCLUDED
3
4 #define LINE_SIZE 120
5 #define FILE_SIZE 30
6 #define STRING_SIZE 20
7 #define TIME_SIZE 15
8 #define COUNT 30
9
10 extern int SHIFTTIME; // func1,func16,option
11 extern int FUNC16; // func16,option
12 extern double ANGO,POW0; // func12,func13
13 extern char input1[FILE_SIZE]; // func1,func3,func16
14 extern char input2[FILE_SIZE]; // func1,func16
15 extern char output[FILE_SIZE]; // func1,func16
16 extern char shift[FILE_SIZE]; // func1,func16
17
18 #endif // HEAD_INCLUDED
```

option.c

```
1 int SHIFTTIME=1; // パワーログの時刻の設定の変更
2 // 0のとき :時刻の調整を行わない
3 // 1のとき :時刻の調整を実行時に手動入力で行う
4 // デフォルトは 0
5
6 int FUNC16=0; // func16 の設定を変更
7 // 0のとき :完成ファイルのリネームし、中間ファイルは削除する
8 // 1のとき :完成ファイルのリネームを行わず、中間ファイルを削除しない
9 // デフォルトは 0
```

makefile

```
1 # マクロ定義
2 CC=gcc
3 OBJS=func1.o func2.o func3.o func4.o func5.o func6.o func7.o func8.o func9.o func10.o func11.o func12.o func13.o func14.o func15.o
4 func16.o option.o
5
6 # コンパイル
7 hita.exe: main.c $(OBJS)
8 $(CC) -o $@ -lm main.c $(OBJS)
9
10 # サフィックスルール
11 .c.o:
12 $(CC) -c $<
13
14 # 不要ファイルの削除
15 .PHONY: clean
16 clean:
17 $(RM) *.o
```

main.c

```
1 int main(void){
2
3 func1();
4 func2();
5 func3();
6 func4();
7 func5();
8 func6();
9 func7();
10 func8();
11 func9();
12 func10();
13 func11();
14 func12();
15 func13();
16 func13();
17 func14();
18 func15();
19 func16();
20
21 return 0;
22 }
```

func1.c

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include <string.h>
4 #include "head.h"
5
6 int SHIFTTIME;
7 char input1[FILE_SIZE],input2[FILE_SIZE],output[FILE_SIZE],shift[FILE_SIZE];
8
9
10 int func1(void){
11
12 FILE*fp;
13 FILE*gp;
14
```

```

15 char line[LINE_SIZE],shift_str[STRING_SIZE];
16 double p1,p2,p3,P1,shift_num;
17
18 // (A)入力ファイルの名前の格納
19 printf(" Please enter input file name of angle log > ");
20 scanf("%s",input1); // input1="20100906.ang"
21 printf(" Please enter input file name of power log > ");
22 scanf("%s",input2); // input2="201009061130.txt"
23 printf(" Please enter the output file name > ");
24 scanf("%s",output); // output="20100906"
25
26 // (B)input1 か input2 と output が一致してしまうときはエラー終了
27 if(strcmp(input1, output) == 0){
28     printf(" error > 'the time of observation' and 'input file name of angle log' have the same name\n");
29     exit(1);
30 }
31 if(strcmp(input2, output) == 0){
32     printf(" error > 'the time of observation' and 'input file name of power log' have the same name\n");
33     exit(1);
34 }
35
36
37 // (C-1)SHIFTTIME が 0 のとき(デフォルト)
38 if(SHIFTTIME==0){
39     shift_num=0;
40 }
41
42
43 // (C-2)SHIFTTIME が 1 のとき(入力を手で変更)
44 else if(SHIFTTIME==1){
45     printf(" Please enter the time shift of power log > ");
46     scanf("%lf",&shift_num); // (C-2-a)例:shift_num=-6(パワーログのほうがアングルログよりも 6秒進んでいるとき)
47     sprintf(shift_str,"%1lf",shift_num); // (C-2-b)shift_str=-6(-6は数字) → shift_str="-6"(-6は文字列)
48     strcpy(shift,input2); // (C-2-c)shift="201009061130.txt"
49     strcat(shift,"_"); // (C-2-d)shift="201009061130.txt_"
50     strcat(shift,shift_str); // (C-2-e)shift="201009061130.txt_-6" 時間のズレを補正したパワーログのファイル名とする
51
52     // (C-2-f)input2 を shift_num だけずらしたファイルを shift へ出力
53     if((fp=fopen(input2,"r"))==NULL){ // "201009061130.txt"の読み込み
54         printf(" Can not read the input file name of power log > %s\n",input2);
55         exit(1);
56     }
57     if((gp=fopen(shift,"w"))==NULL){ // "201009061130.txt_-6"という名前のファイルに書き込み
58         printf(" Unable to open output file\n");
59         exit(1);
60     }
61     while(fscanf(fp,"%lf%lf%lf",&p1,&p2,&p3)!=EOF){
62         P1=p1+shift_num; // パワーログの時間をshift_num だけ調整
63         fprintf(gp,"%3lf %.4lf %.4lf\n",P1,p2,p3);
64     }
65     fclose(fp); // ファイルのクローズ
66     fclose(gp);
67 }
68
69 // (C-3)SHIFTTIME が 0 でも 1 でもない場合エラー終了
70 else{
71     printf(" error > SHIFTTIME\n");
72     exit(1);
73 }
74
75 // (D)最終行に-1を追加
76
77 // (D-1)input1="20100906.ang"からfile1-1への出力
78 if((fp=fopen(input1,"r"))==NULL){
79     printf(" Can not read the input file name of angle log > %s\n",input1);
80     exit(1);
81 }
82 if((gp=fopen("file1-1","w"))==NULL){
83     printf(" Unable to open output file\n");
84     exit(1);
85 }
86 while(fgets(line,STRING_SIZE,fp)!=NULL){
87     fprintf(gp,"%s",line); // "20100906.ang"を"file1-1"へ複写
88 }
89 fprintf(gp,"-1\n"); // -1を追加
90 fclose(fp);
91 fclose(gp);
92
93 // (D-2)input2="201009061130.txt"からfile1-2への出力
94 if((fp=fopen(input2,"r"))==NULL){
95     printf(" Can not read the input file name of power log > %s\n",input2);
96     exit(1);
97 }
98 if((gp=fopen("file1-2","w"))==NULL){
99     printf(" Unable to open output file\n");
100     exit(1);
101 }
102 while(fscanf(fp,"%lf%lf%lf",&p1,&p2,&p3)!=EOF){
103     P1=p1+shift_num; // SHIFTTIME が 0 のときは shift_num=0なので変化なし、1のときは入力したshift_num が足される
104     fprintf(gp,"%3lf %.4lf %.4lf\n",P1,p2,p3);
105 }
106 fprintf(gp,"-1\n");
107 fclose(fp);
108 fclose(gp);
109
110 return 0;
111 }

```

func2.c

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include <string.h>
4 #include <math.h>
5 #include "head.h"

```

```

6
7 int func2(void){
8
9 FILE*fp;
10 FILE*gp;
11
12 int i;
13 double a1,A_startAD,A1,A_endAD,A_start,A_end,p1,P_start,P1,P_end,year,day,hour,minute,second,Year,Day,Hour,Minute;
14 char line[LINE_SIZE],str[STRING_SIZE];
15
16
17 // (A)アングルログ (file1-1)の最初と最後の行の数値をそれぞれA_start_AD と A_end_AD に格納
18 if((fp=fopen("file1-1","r"))==NULL){
19 printf(" Can not read the input file\n");
20 exit(1);
21 }
22 for(i=0;;i++){
23     // カウントi を 0 から 1 ずつ増やす(上限なし)
24     fgets(line,LINE_SIZE,fp); // file1-1の文字列 1行をline に格納
25     sscanf(line,"%[0-9-]",str); // 0から 9までの数字とマイナス [-]をstr に読み込み
26     a1=atof(str); // char 型の str を double 型へとキャストし、a1 に格納
27     if(i==0)A_startAD=a1; // (A-1)ループの最初なら一番最初の行の数値をA_start_AD に格納
28     else if(a1!=-1){ // (A-3)格納したa1 の値が-1(最後の行)ならば最後の行の 1つ手前の数値をA_end_AD に格納してループを抜ける
29         A_endAD=A1;
30         break;
31     }
32     else if(i!=0 && a1!=-1)A1=a1; // (A-2)それ以外は更新したa1 を A1 に格納する
33 }
34 fclose(fp);
35
36 // (B)西暦 (2010249013634)→総秒数 (63419506594)
37 // A_startAD → A_start
38 year=floor(A_startAD/pow(10,9)); // (B-1)2010
39 day=floor(A_startAD/pow(10,6))-1000*year; // 249
40 hour=floor(A_startAD/pow(10,4))-pow(10,5)*year-100*day; // 1
41 minute=floor(A_startAD/100)-pow(10,7)*year-pow(10,4)*day-100*hour; // 36
42 second=A_startAD-pow(10,9)*year-pow(10,6)*day-pow(10,4)*hour-100*minute; // 34
43 Day=(year-1)*365+floor((year-1)/4)-floor((year-1)/100)+floor((year-1)/400)+(day-1)+2; // (B-2)総日数 (+2日)
44 Hour=24*Day+hour; // (B-3)総時間
45 Minute=60*Hour+minute; // (B-4)総分数
46 A_start=60*Minute+second; // (B-5)総秒数
47
48 // A_endAD → A_end
49 year=floor(A_endAD/pow(10,9)); // 2010
50 day=floor(A_endAD/pow(10,6))-1000*year; // 249
51 hour=floor(A_endAD/pow(10,4))-pow(10,5)*year-100*day; // 1
52 minute=floor(A_endAD/100)-pow(10,7)*year-pow(10,4)*day-100*hour; // 36
53 second=A_endAD-pow(10,9)*year-pow(10,6)*day-pow(10,4)*hour-100*minute; // 34
54 Day=(year-1)*365+floor((year-1)/4)-floor((year-1)/100)+floor((year-1)/400)+(day-1)+2; // 総日数 (+2日)
55 Hour=24*Day+hour;
56 Minute=60*Hour+minute;
57 A_end=60*Minute+second;
58
59
60 // (C)パワーログ (file1-2)の最初と最後の行の数値をそれぞれP_start と P_end に格納
61 if((gp=fopen("file1-2","r"))==NULL){
62 printf(" Can not read the input file\n");
63 exit(1);
64 }
65 for(i=0;;i++){
66     fscanf(gp,"%lf%*f%*f",&p1);
67     if(i==0)P_start=p1;
68     else if(p1!=-1){
69         P_end=P1;
70         break;
71     }
72     else if(i!=0 && p1!=-1)P1=p1;
73 }
74 fclose(gp);
75
76
77 // (D)アングルログとパワーログの共通時間の判定
78 if(A_end<=P_start || P_end<=A_start){ // (D-1)
79 printf(" angle log > %.0lf - %.0lf (%.0lf - %.0lf)\n",A_start,A_end,A_startAD,A_endAD);
80 printf(" power log > %.3lf %.3lf\n",P_start,P_end);
81 printf(" There is no common time of the two input files\n");
82 exit(1);
83 }
84
85 else if(P_end>A_end){ // (D-2)
86     if(P_start<A_start){ // (D-2-1)
87         printf(" %.0lf - %.0lf (%.0lf - %.0lf) is a common time\n",A_start,A_end,A_startAD,A_endAD);
88     }
89     if(P_start>A_start){ // (D-2-2)
90         printf(" %.3lf - %.0lf(%.0lf) is a common time\n",P_start,A_end,A_endAD);
91     }
92 }
93
94 else if(A_end>P_end){ // (D-3)
95     if(A_start<P_start){ // (D-3-1)
96         printf(" %.3lf - %.3lf is a common time\n",P_start,P_end);
97     }
98     if(A_start>P_start){ // (D-3-2)
99         printf(" %.0lf(%.0lf) - %.3lf is a common time\n",A_start,A_startAD,P_end);
100     }
101 }
102
103 else{ // (D-4)
104     printf(" error\n");
105     exit(1);
106 }
107
108 return 0;
109 }
110

```

func3.c

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include <string.h>

```

```

4  #include "head.h"
5
6  char input1[FILE_SIZE];
7
8  int func3(void){
9
10 FILE*fp;
11 FILE*gp;
12 FILE*hp;
13 FILE*ip;
14
15 int frag=0,len1,len2;
16 double START,END;
17 char line[LINE_SIZE],str2[STRING_SIZE],str1[STRING_SIZE],strSTART[STRING_SIZE];
18 char strA1[STRING_SIZE],strA2[STRING_SIZE],strA3[STRING_SIZE],strA4[STRING_SIZE],strA5[STRING_SIZE];
19 char strA6[STRING_SIZE],strA7[STRING_SIZE],strA8[STRING_SIZE],strA9[STRING_SIZE],strA10[STRING_SIZE];
20
21 if((fp=fopen(input1,"r"))==NULL){
22 printf(" Can not read the input file\n");
23 exit(1);
24 }
25
26 if((gp=fopen("file3-1","w"))==NULL){
27 printf(" Unable to open output file\n");
28 exit(1);
29 }
30
31 while(fgets(line,LINE_SIZE,fp)!=NULL){ // fp の文字列を line に格納
32 sscanf(line,"%[0-9*]",str1); // line から 0 から 9 と*を読み込みstr1 へ格納
33 len1=strlen(str1); // 文字列str1 の文字数を len1 へと格納
34
35 if(len1==14 && len2==14){ // (2)1つ前の行がコメント行 (len2=14)でかつ新たに取得した行もコメント行である (len2=14)とき
36 strcpy(str2,str1); // str1 を str2 にコピーする
37 len2=strlen(str2); // 文字列str2 の文字数を len2 へと格納 (len2=14)
38 }
39 else if(len1==13 && len2==14){ // (3)1つ前の行がコメント (len2=14)でかつ新たに取得した行がデータ行である (len1=13)
40 fprintf(gp,"%s\n",line); // データ行line を gp へ出力(時間以外のデータを使う。時間は後でstrSTART で置き換える
41 START=atof(strSTART); // 最初のコメント行の秒数となるstrSTART をダブル型の START へキャストする
42 END=atof(str1); // 最初のデータ行の秒数となるstr1 をダブル型の END へキャストする
43 // END-START が 0 か 1 のときはコメント行は飛んでいない
44 break; // ループの脱出
45 }
46 else if(len1==14){ // (1)最初の行がコメント*で始まる場合ここから
47 sscanf(str1,"%[0-9]",strSTART); // 最初のコメント行の秒数をstrSTART に格納
48 strcpy(str2,str1); // str1 を str2 にコピーする
49 len2=strlen(str2); // 文字列str2 の文字数を len2 へと格納
50 }
51
52 else{ // (4) 最初の行がコメント行でないとき
53 frag=1;
54 break;
55 }
56 }
57
58 fclose(fp);
59 fclose(gp);
60
61 if((fp=fopen(input1,"r"))==NULL){
62 printf(" Can not read the input file\n");
63 exit(1);
64 }
65
66 if((gp=fopen("file3-1","r"))==NULL){
67 printf(" Can not read the input file\n");
68 exit(1);
69 }
70
71 if((hp=fopen("file3-2","w"))==NULL){
72 printf(" Unable to open output file\n");
73 exit(1);
74 }
75
76 if((ip=fopen("file3-3","w"))==NULL){
77 printf(" Unable to open output file\n");
78 exit(1);
79 }
80
81 if(frag==0 && (END-START)!=0){ // (5) (4)には入らず、かつ、fp の最初の行がコメント行であるか END-START の数値の差が 1 以上のとき
82 fscanf(gp,"%s%s%s%s%s%s%s%s%s",strA1,strA2,strA3,strA4,strA5,strA6,strA7,strA8,strA9,strA10); // gp の最初の 1 行の各列を各変数へ格納
83 fprintf(hp,"%s %s %s %s %s %s %s %s %s %s\n",strSTART,strA2,strA3,strA4,strA5,strA6,strA7,strA8,strA9,strA10);
84 // ↑ 1列目の時間のみstrSTART へと置き換えて hp へ出力
85 fprintf(ip,"%s %s %s %s %s %s %s %s %s %s\n",strSTART,strA2,strA3,strA4,strA5,strA6,strA7,strA8,strA9,strA10);
86 // ↑ 1列目の時間のみstrSTART へと置き換えて hp へ出力
87 }
88
89 while(fgets(line,LINE_SIZE,fp)!=NULL){ // hp と ip の 2 行目以降に 20100906.ang の 1 行目以降をを複写
90 fprintf(hp,"%s",line);
91 fprintf(ip,"%s",line);
92 }
93
94 fprintf(hp,"1"); // hp の最後の行に 1 を追加(func4.c のための準備)
95
96 fclose(fp);
97 fclose(gp);
98 fclose(hp);
99 fclose(ip);
100
101 return 0;
102 }

```

func4.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include <string.h>
4  #include "head.h"
5
6  int func4(void){
7
8  FILE*fp;
9  FILE*gp;

```

```

10 unsigned int len1,len2;
11 double START,END;
12 char line1[LINE_SIZE],line2[LINE_SIZE],str2[STRING_SIZE],str1[STRING_SIZE],strSTART[STRING_SIZE],strEND[STRING_SIZE];
13 char strA1[STRING_SIZE],strA2[STRING_SIZE],strA3[STRING_SIZE],strA4[STRING_SIZE],strA5[STRING_SIZE];
14 char strA6[STRING_SIZE],strA7[STRING_SIZE],strA8[STRING_SIZE],strA9[STRING_SIZE],strA10[STRING_SIZE];
15 char *old="file3-3",*new="file4-1";
16
17 if((fp=fopen("file3-2","r"))==NULL){
18     printf(" Can not read the input file\n");
19     exit(1);
20 }
21
22
23 if((gp=fopen("file3-3","a"))==NULL){ // 追加書き込み
24     printf(" Unable to open output file\n");
25     exit(1);
26 }
27
28 while(fgets(line1,LINE_SIZE,fp)!=NULL){
29     sscanf(line1,"%[0-9*]",str1);
30     len1=strlen(str1);
31     if(len1==14 && len2==13 || len1==14 && len2==14){ // (2) 13行から 14行または 14行から 14行へと移動するとき
32         strcpy(str2,str1); // strcpy(line2,line1)はせず、14行のときのline2 の情報は持たない(13行のときのline2 の情報を保持)
33         len2=strlen(str2);
34     }
35     else if(len1==1 && len2==14){ // (3) 最後の行がコメントで終わるとき
36         sscanf(str2,"%[0-9]",strEND);
37         START=atof(strSTART);
38         END=atof(strEND);
39         if((END-START)!=0){ // (5) // double 型へキャスト
40             sscanf(line2,"%s%s%s%s%s%s%s%s%s",strA1,strA2,strA3,strA4,strA5,strA6,strA7,strA8,strA9,strA10);
41             fprintf(gp,"%s %s %s %s %s %s %s %s %s %s\n",strEND,strA2,strA3,strA4,strA5,strA6,strA7,strA8,strA9,strA10); // 追
42             // 加書き込み
43         }
44         break; // ループ脱出
45     }
46     else if( (len1==13||len1==14) || (len1==13&&(len2==13||len2==14)) ){ // (1) 最初の行または
47         // 13行か 14行から 13行へと移動するとき
48         strcpy(str2,str1);
49         strcpy(line2,line1); // len1=13かつlen2=14になったらline2 の更新の再開
50         sscanf(line2,"%[0-9]",strSTART); // line2 の 1 列目の時間を strSTART へ格納
51         len2=strlen(str2);
52     }
53     else if(len1==1 && len2==13)break; // (4)最後の行がコメントで終わらないとき
54 }
55 fclose(fp);
56 fclose(gp);
57
58 // file3-3から 4-1へリネーム
59 if(rename(old,new)!=0){
60     printf( " %s from %s failed name change\n",old,new);
61 }
62 else rename(old,new);
63 return 0;
64 }

```

func5.c

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include <string.h>
4 #include "head.h"
5
6 int func5(void)
7 {
8
9     FILE*fp;
10    FILE*gp;
11    FILE*hp;
12
13    int i;
14    unsigned int len;
15    char line[LINE_SIZE],str[STRING_SIZE];
16
17    if((fp=fopen("file4-1","r"))==NULL){
18        printf(" Can not read the input file\n");
19        exit(1);
20    }
21
22    if((gp=fopen("file5-1","w"))==NULL){
23        printf(" Unable to open output file\n");
24        exit(1);
25    }
26
27    if((hp=fopen("file5-2","w"))==NULL){
28        printf(" Unable to open output file\n");
29        exit(1);
30    }
31
32    while(fgets(line,LINE_SIZE,fp)!=NULL){ // (1)file4-1を毎行line へと格納
33        sscanf(line,"%[0-9*]",str); // (2)1列目 (0から 9と*のみ)をスキャンしstr へ格納
34        len=strlen(str); // (3)srt の文字数を len に格納
35        if(len==13){ // (4-a)文字数len が 13(データ行)なら
36            fprintf(gp,"%s",line);
37        }
38        else if(len==14){ // (4-b)文字数len が 14(コメント行)なら
39            fprintf(hp,"%s",line);
40        }
41    }
42
43    fprintf(gp,"-1");
44
45    fclose(fp);
46    fclose(gp);
47    fclose(hp);
48
49    return 0;
50
51 }

```

func6.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include <math.h>
4
5  int func6(void){
6
7  FILE*fp;
8  FILE*gp;
9  int i;
10 double a1,a1_AD,a2,a3,a4,a5,a6,a7,a8,a9,a10,year,day,hour,minute,second,Year,Day,Hour,Minute;
11
12 if((fp=fopen("file5-1","r"))==NULL){
13 printf(" Can not read the input file\n");
14 exit(1);
15 }
16
17 if((gp=fopen("file6-1","w"))==NULL){
18 printf(" Unable to open output file\n");
19 exit(1);
20 }
21
22 // 西暦時間→経過した総秒数
23 for(i=0;;i++){
24 fscanf(fp,"%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",&a1_AD,&a2,&a3,&a4,&a5,&a6,&a7,&a8,&a9,&a10);
25 if(a1_AD==1)break;
26 else{
27     year=floor(a1_AD/pow(10,9)); // 2010
28     day=floor(a1_AD/pow(10,6))-1000*year; // 249
29     hour=floor(a1_AD/pow(10,4))-pow(10,5)*year-100*day; // 1
30     minute=floor(a1_AD/100)-pow(10,7)*year-pow(10,4)*day-100*hour; // 36
31     second=a1_AD-pow(10,9)*year-pow(10,6)*day-pow(10,4)*hour-100*minute; // 34
32     Day=(year-1)*365+floor((year-1)/4)-floor((year-1)/100)+floor((year-1)/400)+(day-1)+2; // 総日数 (+2日)
33     Hour=24*Day+hour;
34     Minute=60*Hour+minute;
35     a1=60*Minute+second;
36     fprintf(gp,"%01f %01f %01f %01f %01f %01f %01f %01f %01f %01f",a1,a1_AD,a2,a3,a4,a5,a6,a7,a8,a9,a10);
37 }
38 }
39
40 fprintf(gp,"-1\n");
41
42 fclose(fp);
43 fclose(gp);
44 return 0;
45 }

```

func7.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include "head.h"
4
5  int func7(void){
6
7  FILE*fp;
8  FILE*gp;
9  FILE*hp;
10
11 int i,j,k;
12 double a[COUNT],A[COUNT];
13
14
15 if((fp=fopen("file6-1","r"))==NULL){
16 printf(" Can not read the input file\n");
17 exit(1);
18 }
19
20 if((gp=fopen("file7-1","w"))==NULL){
21 printf(" Unable to open output file\n");
22 exit(1);
23 }
24
25 if((hp=fopen("file7-2","w"))==NULL){
26 printf(" Unable to open output file\n");
27 exit(1);
28 }
29
30 for(i=0;;i++){
31     if(i==0){ // (1)i=0のとき (最初の行のとき)
32         fscanf(fp,"%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",&a[1],&a[2],&a[3],&a[4],&a[5],&a[6],&a[7],&a[8],&a[9],&a[10],&a[11]);
33         for(k=1;k<=11;k++){
34             A[k]=a[k]; // 1行目の値を格納する
35         }
36     }
37     else if(i!=0){ // i=0でないとき (最初の行でないとき)
38         fscanf(fp,"%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",&a[1],&a[2],&a[3],&a[4],&a[5],&a[6],&a[7],&a[8],&a[9],&a[10],&a[11]);
39         if(a[1]==-1){ // (4)1列目に格納するa[1]の値が-1のとき
40             fprintf(gp,"%01f %01f %01f %01f %01f %01f %01f %01f %01f %01f %01f",A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11]);
41             break; // 最後に更新した最終行をgp へ出力してループを脱出
42         }
43         else if(a[1]-A[1]==1){ // (2)差が1秒のとき (通常)
44             fprintf(gp,"%01f %01f %01f %01f %01f %01f %01f %01f %01f %01f %01f",A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11]);
45             for(k=1;k<=11;k++){
46                 A[k]=a[k]; // 格納する値の更新
47             }
48         }
49         else if(a[1]-A[1]!=1){ // (3) 差が1秒でないとき (時間が抜けているとき)
50             fprintf(hp,"%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf",A[2],a[2],A[1],a[1]); // 抜け始めA[2](A[1])と抜け終わりa[2](a[1])をhp へ出力
51             for(j=0;j<a[1]-A[1];j++){ // 間を (a[1]-A[1]-1)行だけ行数を抜け始めのアンクル情報で補完
52                 fprintf(gp,"%01f %01f %01f %01f %01f %01f %01f %01f %01f %01f %01f",A[1],A[2],A[3],A[4],A[5],A[6],A[7],A[8],A[9],A[10],A[11]);
53             }
54             for(k=1;k<=11;k++){
55                 A[k]=a[k]; // 格納する値の更新
56             }
57         }
58     }
59 }

```

```

59 }
60
61 fprintf(gp, "-1\n");
62 fprintf(hp, "-1\n");
63
64 fclose(fp);
65 fclose(gp);
66 fclose(hp);
67
68 return 0;
69 }

```

func8.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include <string.h>
4  #include "head.h"
5
6  int func8(void){
7      FILE*fp;
8      FILE*gp;
9
10
11      unsigned int len1,len2;
12      char line[LINE_SIZE],str2[STRING_SIZE],str1[STRING_SIZE];
13
14      if((fp=fopen("file4-1","r"))==NULL){
15          printf(" Can not read the input file\n");
16          exit(1);
17      }
18
19      if((gp=fopen("file8-1","w"))==NULL){
20          printf(" Unable to open output file\n");
21          exit(1);
22      }
23
24      while(fgets(line,LINE_SIZE,fp)!=NULL){
25          sscanf(line,"%[0-9*]",str1);
26          len1=strlen(str1);
27          if(len1==14 && len2==13){ // (2)
28              fprintf(gp,"%s\n",str2); // 抜け始めの時間を出力（奇数行）
29              strcpy(str2,str1);
30              len2=strlen(str2);
31          }
32          else if(len1==14 && len2==14){ // (3)
33              strcpy(str2,str1);
34              len2=strlen(str2);
35          }
36          else if(len1==13 && len2==14){ // (4)
37              fprintf(gp,"%s\n",str1); // 抜け終わりを出力
38              strcpy(str2,str1);
39              len2=strlen(str2);
40          }
41          else if(len1==13){ // (1)
42              strcpy(str2,str1);
43              len2=strlen(str2);
44          }
45      }
46
47      fprintf(gp, "-1\n");
48
49      fclose(fp);
50      fclose(gp);
51
52      return 0;
53  }

```

func9.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int func9(void){
5      FILE*fp;
6      FILE*gp;
7
8      int i;
9      double a,A1,A2;
10
11      if((fp=fopen("file8-1","r"))==NULL){
12          printf(" Can not read the input file\n");
13          exit(1);
14      }
15
16
17      if((gp=fopen("file9-1","w"))==NULL){
18          printf(" Unable to open output file\n");
19          exit(1);
20      }
21
22      for(i=1;;i++){ // i=1からスタート
23          fscanf(fp,"%lf",&a);
24          if(a==-1)break;
25          else{
26              if(i%2==1){ // (1)奇数のとき
27                  A1=a;
28              }
29              else if(i%2==0){ // (2)偶数のとき
30                  A2=a;
31                  fprintf(gp,"%01f %01f\n",A1,A2);
32              }
33          }
34      }
35
36      fprintf(gp, "-1");
37
38      fclose(fp);
39      fclose(gp);

```

```

40 return 0;
41
42 }
43

```

func10.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int func10(void){
5
6  FILE*fp;
7  FILE*gp;
8  FILE*hp;
9  FILE*ip;
10
11  int i,j;
12  double x1,x2,x3,x4,y1,y2;
13
14  if((fp=fopen("file7-2","r"))==NULL){
15  printf(" Can not read the input file\n");
16  exit(1);
17  }
18
19  if((hp=fopen("file10-1","w"))==NULL){
20  printf(" Unable to open output file\n");
21  exit(1);
22  }
23
24  if((ip=fopen("file10-2","w"))==NULL){
25  printf(" Unable to open output file\n");
26  exit(1);
27  }
28
29  for(i=0;;i++){
30  fscanf(fp,"%lf%lf%lf%lf",&x1,&x2,&x3,&x4);
31  if(x1==-1)break; // (6)
32  for(j=0;;j++){
33  if(j==0){ // (1)
34  if((gp=fopen("file9-1","r"))==NULL){
35  printf(" Can not read the input file\n");
36  exit(1);
37  }
38  }
39  fscanf(gp,"%lf%lf",&y1,&y2);
40  if(x1>y1){ // (3)
41  if(y1==-1){ // (5)
42  fprintf(ip,"%01f %01f %01f %01f\n",x1,x2,x3,x4);
43  fclose(gp);
44  break;
45  }
46  }
47  }
48  else if(x1==y1){ // (2)
49  fprintf(hp,"%01f %01f %01f %01f\n",x1,x2,x3,x4);
50  fclose(gp);
51  break;
52  }
53  else if(x1<y1){ // (4)
54  fprintf(ip,"%01f %01f %01f %01f\n",x1,x2,x3,x4);
55  fclose(gp);
56  break;
57  }
58  }
59  }
60
61  fclose(fp);
62  fclose(hp);
63  fclose(ip);
64
65  return 0;
66  }

```

func11.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include <string.h>
4  #include <math.h>
5  #include "head.h"
6
7  int func11(void)
8  {
9
10 FILE*fp;
11 FILE*gp;
12
13 int i,t1,t2,t3,t4,t5,second,minute,hour,day,year;
14 double a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,A1,A0;
15 char T1[TIME_SIZE],T2[TIME_SIZE],T3[TIME_SIZE],T4[TIME_SIZE],T5[TIME_SIZE];
16 char T01[TIME_SIZE]="0",T02[TIME_SIZE]="0",T03[TIME_SIZE]="0",T04[TIME_SIZE]="0",T004[TIME_SIZE]="00";
17
18 if((fp=fopen("file7-1","r"))==NULL){
19 printf(" Can not read the input file\n");
20 exit(1);
21 }
22
23 if((gp=fopen("file11-1","w"))==NULL){
24 printf(" Unable to open output file\n");
25 exit(1);
26 }
27
28
29 for(i=0;;i++){
30 fscanf(fp,"%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",&a1,&a2,&a3,&a4,&a5,&a6,&a7,&a8,&a9,&a10,&a11);
31 if(a1==-1)break;
32
33
34 // (A)繰り上がりを考慮した正しい数値をint 型で取得(2010000235959を例に扱う)

```

```

35 if(i==0){
36     A0=a1;          // 1列目の総秒数をA0へ格納 例)2010000235959のときはA0=63398073599
37     year=floor(a2/pow(10,9));
38     day=floor(a2/pow(10,6))-1000*year;          // 2010
39     hour=floor(a2/pow(10,4))-pow(10,5)*year-100*day;          // 0
40     minute=floor(a2/100)-pow(10,7)*year-pow(10,4)*day-100*hour;          // 23
41     second=a2-pow(10,9)*year-pow(10,6)*day-pow(10,4)*hour-100*minute;          // 59
42 }
43
44 A1=A0+i;          // iを利用することで1列目の総秒数A0は1行ごとに+1をすることができる
45 // 例)1行目(i=0)はA0=63398073599+0=63398073599、2行目(i=1)はA0=63398073599+1=63398073600、3行目は..
46 // しかし2列目の西暦秒数は10進数、365(366)進数、24進数、60進数、60進数となっているため総秒数のように単純に+1すればよいというわけにはいかない
47
48 // 最初の行のときは、t1,t2,t3,t4,t5にそれぞれ初期値を格納
49 if(i==0){
50     t1=second+i;          // t1=59+0
51     t2=minute;          // t2=1
52     t3=hour;          // t3=23
53     t4=day;          // t4=59
54     t5=year;          // t5=59
55 }
56
57 // 2行目以降は、基本的に秒数t1にカウントiを足すことで毎行1秒ずつ加算していく
58 else if(i!=0){          // 最初の行でない場合は1つ前の行で格納していた秒数+1をt1に格納(更新)
59     t1=t1+1;          // 59+1(i=1である2行目のとき)=60
60 }
61
62 // 秒数t1は60進数なので59→60になったときは60→0へとリセットし、分数t2を+1繰り上げないといけない。分数
63 // t2、時間数t3においても同じ処理を行う
64 if(t1==60){          // 秒数t1が60になるとき
65     t1=0;          // 秒数t1を60→0へ
66     t2=t2+1;          // その代わりに分数t2を+1(t2=59+1=60)
67 }
68 if(t2==60){          // 上で秒数t1を60→0とし、その代わりに分数t2が59→60となったとき
69     t2=0;          // 分数t2を60→0へ
70     t3=t3+1;          // その代わりに時間数t3を+1(t3=23+1=24)
71 }
72 if(t3==24){          // 上で秒数t1を60→0とし、その代わりに分数t2が59→60とし、更にその代わりに時間数t3が23→24となったとき
73     t3=0;          // 時間数t3を24→0へ
74     t4=t4+1;          // 日数t4を0→1
75     // day以降は考えない(365235959+1と366235959+1の2パターンがあり条件分岐がややこしくなるため)
76 }
77
78 // (B)取得したint型の数値を1ではなく01という書式に変換(以降は2010001010101(t1=1,t2=1,t3=t4=1,t5=2010)を例に扱う)
79 // 2010001010101のような数字の場合(1)の処理ではint型の数値t5=2010,t4=1,t3=1,t2=1,t1=1を取得する
80 // int型で01のような表記にすることはできないため、このままでは西暦時間のような書式を表現することができない
81 // そこで以下では取得した数値を一旦文字列化して、"0"または"00"を文字として結合することで01のような表記ができるようにしている
82
83 // 格納した数値の文字列化
84 sprintf(T1,"%d",t1);          // int型の数値t1=1をchar型の文字列T1="1"へと変換
85 sprintf(T2,"%d",t2);          // t2=1→T2="1"
86 sprintf(T3,"%d",t3);          // t3=1→T3="1"
87 sprintf(T4,"%d",t4);          // t4=1→T4="1"
88 sprintf(T5,"%d",t5);          // t5=2010→T5="2010"
89
90 // 文字列"0"または"00"の追加
91 if(strlen(T1)==1){          // T1の文字数が1文字、つまりT1="0"-"9"までの1桁の値であるとき
92     strcat(T01,T1);          // 文字列T01="0"に文字列T1="1"を結合 T01="01"
93 }
94 if(strlen(T2)==1){          // T2が"0"-"9"のとき
95     strcat(T02,T2);          // T02="0"にT2="1"を結合 T02="01"
96 }
97 if(strlen(T3)==1){          // T3が"0"-"9"のとき
98     strcat(T03,T3);          // T03="0"にT3="1"を結合 T03="01"
99 }
100 if(strlen(T4)==1){          // T4が"0"-"9"のとき
101     strcat(T004,T4);          // T004="00"にT4="1"を結合 T004="001"
102 }
103 else if(strlen(T4)==2){          // T4が"10"-"99"のとき
104     strcat(T04,T4);          // 例)T4=10のとき:T04="0"にT4=10を結合 T04="010"
105 }
106
107 // 結合と文字列T01、T02、T03、T04、T004の初期化
108 if(strlen(T4)==1){          // T4が"0"-"9"のとき
109     strcat(T5,T004);          // T5="2010"にT004="001"を結合 T5="2010001"
110     strcpy(T004,"00");          // 次に備えてT004="00"に初期化
111 }
112 else if(strlen(T4)==2){          // T4が"10"-"99"のとき
113     strcat(T5,T04);          // 例)T4=10でT04="010"のとき:T5="2010"にT04="010"を結合 T5="2010010"
114     strcpy(T04,"0");          // T04の初期化
115 }
116 else if(strlen(T4)==3){          // T4が"100"-"366"のとき
117     strcat(T5,T4);          // 例)T4="100"のとき:T5="2010"にT4="100"を結合 T5="2010100"
118 }
119
120 if(strlen(T3)==1){          // T3が"0"-"9"のとき
121     strcat(T5,T03);          // T5="2010001"にT03="01"を結合 T5="201000101"
122     strcpy(T03,"0");          // T03の初期化
123 }
124 else if(strlen(T3)==2){          // T3が"10"-"23"のとき
125     strcat(T5,T3);          // 例)T3="10"のとき:T5="2010001"にT3="10"を結合 T5="201000110"
126 }
127
128 if(strlen(T2)==1){          // T2が"0"-"9"のとき
129     strcat(T5,T02);          // T5="201000101"にT02="01"を結合 T5="20100010101"
130     strcpy(T02,"0");          // T02の初期化
131 }
132 else if(strlen(T2)==2){          // T2が"10"-"60"のとき
133     strcat(T5,T2);          // 例)T2="10"のとき:T5="201000101"にT2="10"を結合 T5="2010000110"
134 }
135
136 if(strlen(T1)==1){          // T1が"0"-"9"のとき
137     strcat(T5,T01);          // T5="20100010101"にT01="01"を結合 T5="2010001010101"
138     strcpy(T01,"0");          // T01の初期化
139 }
140 else if(strlen(T1)==2){          // T1が"10"-"60"のとき

```



```

95 return 0;
96
97 }
98

```

func13.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include "head.h"
4
5  double ANGO,POWO;
6
7  int func13(void){
8
9  FILE*fp;
10 FILE*gp;
11 FILE*hp;
12
13 int i,j,k;
14 double A[COUNT],a[COUNT],A_2,p1,p2,p3,A_before[COUNT],A_after[COUNT];
15 A_after[1]=0;
16
17 // 読み込むファイルの組み合わせ選択
18 if(ANGO>=POWO){ // (a)扱いにくい時間帯を除去したパワーログ (file12-1)、アングルログ (file11-1)の組み合わせ
19
20     if((fp=fopen("file12-1","r"))==NULL){
21         printf(" Can not read the input file\n");
22         exit(1);
23     }
24     if((gp=fopen("file11-1","r"))==NULL){
25         printf(" Can not read the input file\n");
26         exit(1);
27     }
28     if((hp=fopen("file13-1","w"))==NULL){
29         printf(" Unable to open output file\n");
30         exit(1);
31     }
32 }
33
34 else if(ANGO<POWO){ // (b)扱いにくい時間帯を除去したアングルログ (file12-1)、パワーログ (file1-2)の組み合わせ
35
36     if((gp=fopen("file12-1","r"))==NULL){
37         printf(" Can not read the input file\n");
38         exit(1);
39     }
40     if((fp=fopen("file1-2","r"))==NULL){
41         printf(" Can not read the input file\n");
42         exit(1);
43     }
44     if((hp=fopen("file13-1","w"))==NULL){
45         printf(" Unable to open output file\n");
46         exit(1);
47     }
48 }
49
50 }
51
52 for(i=0;;i++){
53
54     fscanf(fp,"%lf%lf%lf",&p1,&p2,&p3); // p1=1.2→1.6:(3)へ
55                                         // p1=1.9→2.2:(4)へ
56                                         // p1=2.5→5.4:(1)→(1)→(2)へ
57
58     if(p1==-1)break; // (6-a)
59
60     for(j=0;;j++){
61
62         if(p1<A_after[1]){ // (3)
63             if(A_after[2]-A_before[2]==41){
64                 A_after[2]=A_after[2]-40;
65             }
66             if(A_after[2]-A_before[2]==4041){
67                 A_after[2]=A_after[2]-4040;
68             }
69             if(A_after[2]-A_before[2]==764041){
70                 A_after[2]=A_after[2]-764040;
71             }
72             for(k=2;k<=11;k++){
73                 A[k]=(A_before[k]*(A_after[1]-p1)+A_after[k]*(p1-A_before[1]))/(A_after[1]-A_before[1]);
74             }
75             fprintf(hp,"%3lf %3lf %4lf %4lf %3lf %3lf %4lf %4lf %4lf %4lf %3lf %3lf %3lf\n",A[2],p1,p2,p3,A[3],A[4],A[5],A[6],A
76                 [7],A[8],A[9],A[10],A[11]);
77             A_after[2]=A_2;
78             break;
79         }
80         else if(p1>=A_after[1]){ // (A)
81             fscanf(gp,"%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",&a[1],&a[2],&a[3],&a[4],&a[5],&a[6],&a[7],&a[8],&a[9],&a[10],&a[11]);
82
83             if(a[1]==-1)goto BREAK; // (6-b)
84
85             if(p1>=a[1]){ // (1)
86                 for(k=1;k<=11;k++){
87                     A_before[k]=a[k];
88                 }
89             }
90
91             else if(p1<a[1]){ // (B)
92
93                 if(a[1]-A_after[1]==1){ // (4)
94                     for(k=1;k<=11;k++){
95                         A_before[k]=A_after[k];
96                         A_after[k]=a[k];
97                     }
98                     if(k==2){
99                         A_2=a[2];
100                     }
101                 }
102                 if(A_after[2]-A_before[2]==41){ // (5)
103                     A_after[2]=A_after[2]-40;
104                 }
105                 if(A_after[2]-A_before[2]==4041){

```

```

105     A_after[2]=A_after[2]-4040;
106 }
107 if(A_after[2]-A_before[2]==764041){
108     A_after[2]=A_after[2]-764040;
109 }
110 for(k=2;k<=11;k++){
111     A[k]=(A_before[k]*(A_after[1]-p1)+A_after[k]*(p1-A_before[1]))/(A_after[1]-A_before[1]);
112 }
113 fprintf(fp,"%0.31f %0.31f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.41f %0.31f %0.31f %0.31f\n",A[2],p1,p2,p3,A[3],A[4],A[5],A
114         [6],A[7],A[8],A[9],A[10],A[11]);
115 A_after[2]=A_2;
116 break;
117 }
118 else if(a[1]-A_after[1]!=1){ // (2)
119     for(k=1;k<=11;k++){
120         A_after[k]=a[k];
121         if(k==2){
122             A_2=a[2];
123         }
124     }
125     if(A_after[2]-A_before[2]==41){
126         A_after[2]=A_after[2]-40;
127     }
128     if(A_after[2]-A_before[2]==4041){
129         A_after[2]=A_after[2]-4040;
130     }
131     if(A_after[2]-A_before[2]==764041){
132         A_after[2]=A_after[2]-764040;
133     }
134     for(k=2;k<=11;k++){
135         A[k]=(A_before[k]*(A_after[1]-p1)+A_after[k]*(p1-A_before[1]))/(A_after[1]-A_before[1]);
136     }
137     fprintf(fp,"%0.31f %0.31f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.41f %0.31f %0.31f %0.31f\n",A[2],p1,p2,p3,A[3],A[4],A[5],A
138         [6],A[7],A[8],A[9],A[10],A[11]);
139     A_after[2]=A_2;
140     break;
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 BREAK;
149
150 fprintf(fp,"-1");
151
152
153 fclose(fp);
154 fclose(gp);
155 fclose(hp);
156
157 return 0;
158
159 }

```

func14.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<math.h>
4
5  int func14(void){
6
7  FILE*fp;
8  FILE*gp;
9  FILE*hp;
10
11  int i,j,loop_count=1;
12  double start_frag=0,P2,P3,p1,p2,p3,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,P1_before,P2_before,P3_before,P1_after,P2_after,P3_after;
13
14
15  if((gp=fopen("file11-1","r"))==NULL){
16  printf(" Can not read the input file\n");
17  exit(1);
18  }
19
20  if((fp=fopen("file1-2","r"))==NULL){
21  printf(" Can not read the input file\n");
22  exit(1);
23  }
24
25  if((hp=fopen("file14-1","w"))==NULL){
26  printf(" Unable to open output file\n");
27  exit(1);
28  }
29
30  for(i=0;;i++){
31  fscanf(gp,"%lf%lf%lf%lf%lf%lf%lf%lf%lf%lf",&a1,&a2,&a3,&a4,&a5,&a6,&a7,&a8,&a9,&a10,&a11);
32
33  if(a1==-1)break; // (7-a)
34
35
36  if(start_frag==1 && a1<=p1)goto J_BREAK; // (3) a1>p1 となるところまで a1 を更新し続ける
37  else if(start_frag==1 && a1>p1)start_frag=2; // (4) フラグの変更
38
39  for(j=0;;j++){
40  if(loop_count==1 && start_frag!=2){ // (1) 指定しないとelse if(loop_count!=1)におけるbreak でスキャンだけ進んでしまう
41  fscanf(fp,"%lf%lf%lf",&p1,&p2,&p3); // すでに一回p1の1行目はスキャン済みなので start_frag!=2で回避
42  if(p1==-1)goto I_BREAK; // (7-b)
43  }
44
45
46  if(i==0 && j==0 && a1<=p1){ // (2)
47  start_frag=1;
48  goto J_BREAK;
49  }
50
51  start_frag=0; // フラグの解除 (start_frag=2→start_frag=0)
52
53  if(loop_count==1){ // (A)

```



```

60         if(det1==--1){           // (D)file13-1のほうがfile14-1よりも先に終わるとき
61             rough0=rough2-t0;
62             fprintf(hp,"%0.31f %0.31f %0.31f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.31f %0.31f\n",rough0,rough1,rough2,rough3,
63                 rough4,rough5,rough6,rough7,rough8,rough9,rough10,rough11,rough12,rough13);
64             goto BREAK;
65         }
66
67         if(rough2>=det2){         // (A) file13-1の出力
68             if(i==0 && j==0){     // (2-a)file13-1のほうがfile14-1よりも開始時間が早いとき
69                 t0=floor(det2);
70             }                     // j++
71             det0=det2-t0;
72             fprintf(hp,"%0.31f %0.31f %0.31f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.31f %0.31f %0.31f\n",det0,det1,det2,det3,det4,det5,
73                 det6,det7,det8,det9,det10,det11,det12,det13);
74         }
75         else if(rough2<det2){     // (B)file14-1の出力
76             if(i==0 && j==0){     // (2-b)file14-1のほうがfile13-1よりも開始時間が早いとき
77                 t0=rough2;
78             }
79             rough0=rough2-t0;
80             fprintf(hp,"%0.31f %0.31f %0.31f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.31f %0.31f %0.31f\n",rough0,rough1,rough2,rough3,
81                 rough4,rough5,rough6,rough7,rough8,rough9,rough10,rough11,rough12,rough13);
82             break;               // 脱出
83         }
84     }
85     ROOP:;
86 }
87
88 fclose(fp);
89 fclose(gp);
90 fclose(hp);
91
92
93 // (B)file13-1の秒数から (A)で取得したt0 を引き、file15-2の 1列目に追加する
94
95 if((fp=fopen("file13-1","r"))==NULL){
96     printf(" Can not read the input file\n");
97     exit(1);
98 }
99
100 if((gp=fopen("file15-2","w"))==NULL){
101     printf(" Unable to open output file\n");
102     exit(1);
103 }
104
105 for(i=0;;i++){
106     fscanf(fp,"%0.31f%0.31f%0.31f%0.41f%0.41f%0.31f%0.31f%0.41f%0.41f%0.41f%0.31f%0.31f%0.31f",&rough1,&rough2,&rough3,&rough4,&rough5,&rough6,&rough7,&rough8,&rough9,&rough10,
107         &rough11,&rough12,&rough13);
108     if(rough1==--1)break;
109     rough0=rough2-t0;
110     fprintf(gp,"%0.31f %0.31f %0.31f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.31f %0.31f %0.31f\n",rough0,rough1,rough2,rough3,
111         rough4,rough5,rough6,rough7,rough8,rough9,rough10,rough11,rough12,rough13);
112 }
113
114 fclose(fp);
115 fclose(gp);
116
117 // (C)file14-1の秒数から (A)で取得したt0 を引き、file15-3の 1列目に追加する
118
119 if((fp=fopen("file14-1","r"))==NULL){
120     printf(" Can not read the input file\n");
121     exit(1);
122 }
123
124 if((gp=fopen("file15-3","w"))==NULL){
125     printf(" Unable to open output file\n");
126     exit(1);
127 }
128
129 for(i=0;;i++){
130     fscanf(fp,"%0.31f%0.31f%0.31f%0.41f%0.41f%0.31f%0.31f%0.41f%0.41f%0.41f%0.31f%0.31f%0.31f",&det1,&det2,&det3,&det4,&det5,&det6,&det7,&det8,&det9,&det10,&det11,&det12,&
131         det13);
132     if(det1==--1)break;
133     det0=det2-t0;
134     fprintf(gp,"%0.01f %0.01f %0.01f %0.41f %0.41f %0.31f %0.31f %0.41f %0.41f %0.41f %0.31f %0.31f %0.31f\n",det0,det1,det2,det3,det4,det5,
135         det6,det7,det8,det9,det10,det11,det12,det13);
136 }
137
138 fclose(fp);
139 fclose(gp);
140
141 // (D)file5-2の秒数から (A)で取得したt0 を引き、file15-4の 1列目に追加する
142
143 if((fp=fopen("file5-2","r"))==NULL){
144     printf(" Can not read the input file\n");
145     exit(1);
146 }
147
148 if((gp=fopen("file15-4","w"))==NULL){
149     printf(" Unable to open output file\n");
150     exit(1);
151 }
152
153 while(fgets(line,LINE_SIZE,fp)!=NULL){
154     sscanf(line,"%[0-9]",str);
155     ref1=atof(str);
156     year=floor(ref1/pow(10,9));
157     day=floor(ref1/pow(10,6))-1000*year;
158     hour=floor(ref1/pow(10,4))-pow(10,5)*year-100*day;
159     minute=floor(ref1/100)-pow(10,7)*year-pow(10,4)*day-100*hour;
160     second=ref1-pow(10,9)*year-pow(10,6)*day-pow(10,4)*hour-100*minute;
161     Day=(year-1)*365+floor((year-1)/4)-floor((year-1)/100)+floor((year-1)/400)+(day-1)+2; // 総日数 (+2日)
162     Hour=24*Day+hour;
163     Minute=60*Hour+minute;
164     ref2=60*Minute+second;
165     ref0=ref2-t0;
166     fprintf(gp,"%0.01f %0.01f %s",ref0,ref2,line);
167 }
168

```

```

167 fclose(fp);
168 fclose(gp);
169
170 return 0;
171 }

```

func16.c

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <sys/stat.h>
5  #include "head.h"
6
7  int SHIFTTIME, FUNC16;
8  char input1[FILE_SIZE], input2[FILE_SIZE], output[FILE_SIZE], shift[FILE_SIZE];
9
10 int func16(void)
11 {
12
13     FILE*fp;
14     FILE*gp;
15
16     int i;
17     char dir[FILE_SIZE];
18     char old[COUNT][FILE_SIZE];
19     char new[COUNT][FILE_SIZE];
20     char del[COUNT][FILE_SIZE];
21     char name1[FILE_SIZE], name2[FILE_SIZE], name3[FILE_SIZE];
22     char line1[LINE_SIZE], line2[LINE_SIZE];
23
24
25     //(A)作成ファイルのリネーム、デリート
26
27     // (A-1)FUNC16 が 0 のとき (完成ファイルのリネームし、中間ファイルを削除するとき)
28     if(FUNC16==0){
29
30         //(A-1-a)ディレクトリ"output/"の作成
31         strcpy(dir, output); // dir に output をコピー (dir="20100906")
32
33         if(mkdir(dir, S_IRUSR | S_IWUSR | S_IXUSR | S_IRGRP | S_IWGRP | S_IXGRP | S_IROTH | S_IXOTH) ==0){
34             printf(" Create a directory for '%s'\n", dir);
35         }
36
37         //(A-1-b)output.○○の形で6つの完成ファイルのリネーム
38         for(i=0; i<=5; i++){
39
40             strcpy(new[i], output); // new[i]にoutput をコピー (new[i]="20100906")
41             strcat(new[i], "/"); // new[i]に"/"を結合 (new[i]="20100906/")
42             strcat(new[i], output); // new[i]にoutput をコピー (new[i]="20100906/20100906")
43
44             if(i==0){
45                 strcpy(old[i], "file10-1"); // old[0]にfile10-1をコピー (old[0]="file10-1")
46                 strcat(new[i], ".com"); // new[0]に".com"を結合 (new[0]="20100906/20100906.com")
47             }
48             else if(i==1){
49                 strcpy(old[i], "file10-2");
50                 strcat(new[i], ".bug");
51             }
52             else if(i==2){
53                 strcpy(old[i], "file15-1");
54                 strcat(new[i], ".merge");
55             }
56             else if(i==3){
57                 strcpy(old[i], "file15-2");
58                 strcat(new[i], ".detail");
59             }
60             else if(i==4){
61                 strcpy(old[i], "file15-3");
62                 strcat(new[i], ".rough");
63             }
64             else if(i==5){
65                 strcpy(old[i], "file15-4");
66                 strcat(new[i], ".ref");
67             }
68             if(rename(old[i], new[i])!=0){ // リネームに成功したとき
69                 rename(old[i], new[i]); // old[i]をnew[i]へリネーム (ファイルがdir へ移動される)
70             }
71             else if(rename(old[i], new[i])!=0){ // リネームに失敗したとき
72                 printf(" %s from %s failed name change\n", old[i], new[i]); // リネームに失敗したと表示する
73             }
74         }
75
76
77         //(A-1-c)それ以外の16個の中間ファイルの削除
78         for(i=0; i<=15; i++){
79
80             if(i==0){
81                 strcpy(del[i], "file1-1"); // del[0]にfile1-1をコピー (del[0]="file1-1")
82             }
83             else if(i==1){
84                 strcpy(del[i], "file1-2");
85             }
86             else if(i==2){
87                 strcpy(del[i], "file3-1");
88             }
89             else if(i==3){
90                 strcpy(del[i], "file3-2");
91             }
92             else if(i==4){
93                 strcpy(del[i], "file4-1");
94             }
95             else if(i==5){
96                 strcpy(del[i], "file5-1");
97             }
98             else if(i==6){
99                 strcpy(del[i], "file5-2");
100             }
101             else if(i==7){
102                 strcpy(del[i], "file6-1");

```

```

103 }
104 else if(i==8){
105     strcpy(del[i], "file7-1");
106 }
107 else if(i==9){
108     strcpy(del[i], "file7-2");
109 }
110 else if(i==10){
111     strcpy(del[i], "file8-1");
112 }
113 else if(i==11){
114     strcpy(del[i], "file9-1");
115 }
116 else if(i==12){
117     strcpy(del[i], "file11-1");
118 }
119 else if(i==13){
120     strcpy(del[i], "file12-1");
121 }
122 else if(i==14){
123     strcpy(del[i], "file13-1");
124 }
125 else if(i==15){
126     strcpy(del[i], "file14-1");
127 }
128
129 if(remove(del[i])==0){ // デリートに成功したとき
130     remove(del[i]); // 使用済みの中間ファイルdel[i]を削除
131 }
132 else if(remove(del[i])!=0){ // デリートに失敗したとき
133     printf(" Failed to delete file > %s\n", del[i]); // デリートに失敗したと表示する
134 }
135 }
136 printf(" Files created to '%s'\n", dir);
137 }
138
139 // (A-2)FUNC16 が 1 のとき(中間ファイルを残すとき)
140 else if(FUNC16==1){
141 // (A-2-a)ディレクトリ"output_part/"の作成
142     strcpy(dir, output); // dir に output をコピー (dir="20100906")
143     strcat(dir, "_part"); // dir に "_part"を結合 (dir="20100906_part")
144
145     if(mkdir(dir, S_IRUSR | S_IWUSR | S_IXUSR | S_IRGRP | S_IWGRP | S_IXGRP | S_IROTH | S_IXOTH | S_IXOTH) ==0){
146         printf(" Create a directory for '%s'\n", dir);
147     }
148
149 // (A-2-b)全ファイルをディレクトリ"output_part/"へと移動(名前はfile o-o)のまま
150     for(i=0; i<=21; i++){
151         strcpy(new[i], output);
152         strcat(new[i], "_part/"); // new[i]に "_part/"を結合 (new[i]="20100906_part/20100906")
153
154         if(i==0){
155             strcpy(old[i], "file1-1");
156             strcat(new[i], "file1-1"); // new[0]="20100906_part/file1-1"
157         }
158         else if(i==1){
159             strcpy(old[i], "file1-2");
160             strcat(new[i], "file1-2");
161         }
162         else if(i==2){
163             strcpy(old[i], "file3-1");
164             strcat(new[i], "file3-1");
165         }
166         else if(i==3){
167             strcpy(old[i], "file3-2");
168             strcat(new[i], "file3-2");
169         }
170         else if(i==4){
171             strcpy(old[i], "file4-1");
172             strcat(new[i], "file4-1");
173         }
174         else if(i==5){
175             strcpy(old[i], "file5-1");
176             strcat(new[i], "file5-1");
177         }
178         else if(i==6){
179             strcpy(old[i], "file5-2");
180             strcat(new[i], "file5-2");
181         }
182         else if(i==7){
183             strcpy(old[i], "file6-1");
184             strcat(new[i], "file6-1");
185         }
186         else if(i==8){
187             strcpy(old[i], "file7-1");
188             strcat(new[i], "file7-1");
189         }
190         else if(i==9){
191             strcpy(old[i], "file7-2");
192             strcat(new[i], "file7-2");
193         }
194         else if(i==10){
195             strcpy(old[i], "file8-1");
196             strcat(new[i], "file8-1");
197         }
198         else if(i==11){
199             strcpy(old[i], "file9-1");
200             strcat(new[i], "file9-1");
201         }
202         else if(i==12){
203             strcpy(old[i], "file10-1");
204             strcat(new[i], "file10-1");
205         }
206         else if(i==13){
207             strcpy(old[i], "file10-2");
208             strcat(new[i], "file10-2");
209         }
210         else if(i==14){
211             strcpy(old[i], "file11-1");
212         }
213     }
214 }

```

```

215     strcat(new[i], "file11-1");
216 }
217 else if(i==15){
218     strcpy(old[i], "file12-1");
219     strcat(new[i], "file12-1");
220 }
221 else if(i==16){
222     strcpy(old[i], "file13-1");
223     strcat(new[i], "file13-1");
224 }
225 else if(i==17){
226     strcpy(old[i], "file14-1");
227     strcat(new[i], "file14-1");
228 }
229 else if(i==18){
230     strcpy(old[i], "file15-1");
231     strcat(new[i], "file15-1");
232 }
233 else if(i==19){
234     strcpy(old[i], "file15-2");
235     strcat(new[i], "file15-2");
236 }
237 else if(i==20){
238     strcpy(old[i], "file15-3");
239     strcat(new[i], "file15-3");
240 }
241 else if(i==21){
242     strcpy(old[i], "file15-4");
243     strcat(new[i], "file15-4");
244 }
245
246 if(rename(old[i], new[i])==0){
247     rename(old[i], new[i]); // 中間ファイルもまとめてdir("20100906_part")へ移動
248 }
249 else if(rename(old[i], new[i])!=0){
250     printf(" %s from %s failed name change\n", old[i], new[i]);
251 }
252 }
253
254 printf(" Files created to '%s'\n", dir);
255 }
256
257 // (A-3)FUNC16 が 0 でも 1 でもないとき
258 else if(FUNC16!=0 && FUNC16!=1){
259     printf(" error > FUNC16\n");
260     exit(1);
261 }
262
263 // (B)最初の読み込みファイル (アングルログとパワーログ)をdir("20100906/"または"20100906_part"へ複写、移動
264
265 // (B-1)アングルログinput1("20100906.ang")の複写、dir への移動
266
267 strcpy(name1, output);
268 if(FUNC16==0){
269     strcat(name1, "/");
270 }
271 else if(FUNC16==1){
272     strcat(name1, "_part/");
273 }
274
275 strcat(name1, input1); // name1 = "20100906/20100906.ang" または "20100906_part/20100906.ang"
276
277 if((fp=fopen(input1, "r"))==NULL){
278     printf(" Can not read the input file\n");
279     exit(1);
280 }
281 if((gp=fopen(name1, "w"))==NULL){
282     printf(" Unable to open output file\n");
283     exit(1);
284 }
285 while(fgets(line1, LINE_SIZE, fp)!=NULL){
286     fprintf(gp, "%s", line1);
287 }
288 fclose(fp);
289 fclose(gp);
290
291 // (B-2-a)SHIFTTIME=0のとき
292 if(SHIFTTIME==0){
293
294 // パワーログinput2("201009061130.txt")の複写、dir への移動
295
296 strcpy(name2, output);
297 if(FUNC16==0){
298     strcat(name2, "/");
299 }
300 else if(FUNC16==1){
301     strcat(name2, "_part/");
302 }
303
304 strcat(name2, input2); // name2 = "20100906/201009061130.txt" または "20100906_part/201009061130.txt"
305
306 if((fp=fopen(input2, "r"))==NULL){
307     printf(" Can not read the input file\n");
308     exit(1);
309 }
310 if((gp=fopen(name2, "w"))==NULL){
311     printf(" Unable to open output file\n");
312     exit(1);
313 }
314 while(fgets(line2, LINE_SIZE, fp)!=NULL){
315     fprintf(gp, "%s", line2);
316 }
317 fclose(fp);
318 fclose(gp);
319
320 // (B-2-b)SHIFTTIME=1のとき
321 else if(SHIFTTIME==1){
322
323 // 時刻修正したパワーログshift("201009061130.txt-3.0")のdir への移動
324
325 strcpy(name3, output);
326 if(FUNC16==0){
327     strcat(name3, "/");

```

```

327     }
328     else if(FUNC16==1){
329         strcat(name3,"_part/");
330     }
331     strcat(name3,shift); // name3 = "20100906/201009061130.txt-3.0" または "20100906_part/201009061130.txt-3.0"
332
333     if(rename(shift,name3)!=0){
334         printf( " %s from %s failed name change\n",shift,name3);
335     }
336     else rename(shift,name3);
337 }
338
339 return 0;
340 }

```

2 その他のソースファイル

2.1 taka.exe のソースファイル

全てを載せると膨大な量になってしまうため、hita.exe と比べて大きく修正した func1 のみ載せる。

func1.c(@takahagi)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include <string.h>
4 #include "head.h"
5
6 int SHIFTTIME,TAKAHIT;
7 char input1[FILE_SIZE],input2[FILE_SIZE],output[FILE_SIZE],shift[FILE_SIZE];
8
9
10 int func1(void){
11
12 FILE*fp;
13 FILE*gp;
14 FILE*hp;
15
16 unsigned int len;
17 char line[LINE_SIZE],shift_str[STRING_SIZE],str1[STRING_SIZE];
18 char str2[STRING_SIZE],str3[STRING_SIZE],str4[STRING_SIZE],str5[STRING_SIZE],str6[STRING_SIZE];
19 char str7[STRING_SIZE],str8[STRING_SIZE],str9[STRING_SIZE],str10[STRING_SIZE],str11[STRING_SIZE];
20 double p1,p2,p3,P1,shift_num;
21
22 // (A)入力ファイルの名前の格納
23 printf(" Please enter input file name of angle log > ");
24 scanf("%s",input1); // input1="20100906.ang"
25 printf(" Please enter input file name of power log > ");
26 scanf("%s",input2); // input2="201009061130.txt"
27 printf(" Please enter the output file name > ");
28 scanf("%s",output); // output="20100906"
29
30
31 // (B)input1 が input2 と output が一致してしまうときはエラー終了
32 if(strcmp(input1, output) == 0){
33     printf(" error > 'the time of observation' and 'input file name of angle log' have the same name\n");
34     exit(1);
35 }
36 if(strcmp(input2, output) == 0){
37     printf(" error > 'the time of observation' and 'input file name of power log' have the same name\n");
38     exit(1);
39 }
40
41
42 // (C-1)SHIFTTIME が 0 のとき(デフォルト)
43 if(SHIFTTIME==0){
44     shift_num=0;
45 }
46
47
48 // (C-2)SHIFTTIME が 1 のとき(入力を手動で変更)
49 else if(SHIFTTIME==1){
50     printf(" Please enter the time shift of power log > ");
51     scanf("%lf",&shift_num); // (C-2-a)例:shift_num=-6(パワーログのほうがアングルログよりも 6秒進んでいるとき)
52     sprintf(shift_str,"%1lf",shift_num); // (C-2-b)shift_str=-6(-6は数字) → shift_str="-6"(-6は文字列)
53     strcpy(shift,input2); // (C-2-c)shift="201009061130.txt"
54     strcat(shift,"_"); // (C-2-d)shift="201009061130.txt_"
55     strcat(shift,shift_str); // (C-2-e)shift="201009061130.txt_-6" 時間のズレを補正したパワーログのファイル名とする
56
57 // (C-2-f)input2 を shift_num だけずらしたファイルを shift へ出力
58 if((fp=fopen(input2,"r"))==NULL){ // "201009061130.txt"の読み込み
59     printf(" Can not read the input file name of power log > %s\n",input2);
60     exit(1);
61 }
62 if((gp=fopen(shift,"w"))==NULL){ // "201009061130.txt_-6"という名前のファイルに書き込み
63     printf(" Unable to open output file\n");
64     exit(1);
65 }
66 while(fscanf(fp,"%lf%lf%lf",&p1,&p2,&p3)!=EOF){
67     P1=p1+shift_num; // パワーログの時間をshift_num だけ調整
68     fprintf(gp,"%1.3lf %1.4lf %1.4lf\n",P1,p2,p3);
69 }
70 fclose(fp); // ファイルのクローズ
71 fclose(gp);
72 }
73
74 // (C-3)SHIFTTIME が 0 でも 1 でもない場合エラー終了
75 else{
76     printf(" error > SHIFTTIME\n");
77     exit(1);
78 }
79
80 // (D)最終行に-1を追加
81
82 // (D-1)input1="20100906.ang"からfile1-1への出力
83 if((fp=fopen(input1,"r"))==NULL){
84     printf(" Can not read the input file name of angle log > %s\n",input1);
85     exit(1);
86 }
87 if((gp=fopen("file1-3","w"))==NULL){ // 追加 :高萩アングルログのカンマを除外したファイル (-1はつけない)
88     printf(" Unable to open output file\n"); // カンマ除外+最終行に-1
89     exit(1);
90 }
91 if((hp=fopen("file1-1","w"))==NULL){
92     printf(" Unable to open output file\n");
93     exit(1);
94 }
95
96 while(fgets(line,LINE_SIZE,fp)!=NULL){
97     sscanf(line,"%[0-9*:] ",str1);
98     len=strlen(str1); // line の文字数を len に格納
99     if(len==13){ // データ行なら
```

```

98 scanf(line,"%[^,]%*[ ,] %[^,]%*[ ,] %[^,]%*[ ,] %[^,]%*[ ,] %[^,]%*[ ,] %[^,]%*[ ,] %[^,]%*[ ,]");
99 printf(gp,"%s %s %s %s %s\n",str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11); // データ行を出力を
// gpへ出力(コメント行は出力されない)
100 fprintf(hp,"%s %s %s %s %s\n",str1,str2,str3,str4,str5,str6,str7,str8,str9,str10,str11);
101 }
102 else if(len==14){ // コメント行なら
103     fprintf(gp,"%s",line);
104     fprintf(hp,"%s",line);
105 }
106 }
107 }
108 }
109 }
110 fprintf(hp,"-1\n"); // -1を追加
111 
112 fclose(fp);
113 fclose(gp);
114 fclose(hp);
115 
116 // (D-2) input2="201009061130.txt"からfile1-2への出力
117 if((fp=fopen(input2,"r"))==NULL){
118     printf(" Can not read the input file name of power log > %s\n",input2);
119     exit(1);
120 }
121 
122 if((gp=fopen("file1-2","w"))==NULL){
123     printf(" Unable to open output file\n");
124     exit(1);
125 }
126 while(fscanf(fp,"%lf%lf%lf",&p1,&p2,&p3)!=EOF){
127     P1=p1+shift_num; // SHIFTTIME が 0 のときは shift_num=0なので変化なし、1のときは入力したshift_num が足される
128     fprintf(gp,"%-.3lf %.4lf %.4lf\n",P1,p2,p3);
129 }
130 
131 fprintf(gp,"-1\n");
132 
133 fclose(fp);
134 fclose(gp);
135 
136 return 0;
137 }
```

2.2 takahit.exe のソースファイル

こちらも全てを載せると膨大な量になってしまうため、大きく修正した箇所や追加ソースのみ載せる。なお `h_func15` とは `hita.exe` の `func1` という意味で、`t_func15` とは `taka.exe` の `func1` という意味である。また、`taka.exe` や `hita.exe` で使っていた、`option.c` はソースファイルとして使っていない。

head.h(@takahit)

```

1  #ifndef HEAD_INCLUDED
2  #define HEAD_INCLUDED
3
4  #define LINE_SIZE 200
5  #define FILE_SIZE 50
6  #define STRING_SIZE 20
7  #define TIME_SIZE 15
8  #define COUNT 30
9
10 extern int SHIFTTIME;    // func1,func16,option
11 extern int FUNC16;       // func16,option
12 extern int TSYS_FRAG;    // main,plus2_1(追加)
13 extern double ANGO,POWO; // func12,func13
14 extern char input1[FILE_SIZE]; // func1,func3,func16
15 extern char input2[FILE_SIZE]; // func1,func16
16 extern char output[FILE_SIZE]; // func1,func16
17 extern char shift[FILE_SIZE];  // func1,func16
18
19 #endif // HEAD_INCLUDED

```

makefile(@takahit)

```

1 # マクロ定義
2 CC=gcc
3 OBJS=t_func1.o t_func2.o t_func3.o t_func4.o t_func5.o t_func6.o t_func7.o t_func8.o t_func9.o t_func10.o t_func11.o t_func12.o
   t_func13.o t_func14.o t_func15.o t_func16.o h_func1.o h_func2.o h_func3.o h_func4.o h_func5.o h_func6.o h_func7.o h_func8.o
   h_func9.o h_func10.o h_func11.o h_func12.o h_func13.o h_func14.o h_func15.o h_func16.o h_plus1.o t_plus1.o plus1_2.o
   h_plus2_1.o t_plus2_1.o plus3_1.o plus3_2.o plus4_1.o
4
5 # コンパイル
6 takahit.exe: main.c $(OBJS)
7     $(CC) -o $@ -lm main.c $(OBJS)
8
9 # サフィックスルール
10 .c.o:
11     $(CC) -c $<
12
13 # 不要ファイルの削除
14 .PHONY: clean
15 clean:
16     $(RM) *.o

```

main.c(@takahit)

```
1 #include <stdio.h>
2 #include <string.h>
3 #include "head.h"
4
5 int SHIFTTIME,FUNC16,TSYS_FRAG;
6
7 int main(int argc, char **argv){
```

```

8
9 int i,takahagi_frag,cut_frag,pow_frag,check_frag;
10
11 SHIFTTIME=0,FUNC16=0,TSYS_FRAG=0,takahagi_frag=0,cut_frag=0,pow_frag=0,check_frag=0; // デフォルトでは 0
12
13
14 for(i=0; i<argc; i++){
15     if( strcmp(argv[i],"-h")==0){
16         takahagi_frag=0;
17     }
18     if( strcmp(argv[i],"-t")==0){
19         takahagi_frag=1;
20     }
21     if( strcmp(argv[i],"-s")==0){
22         SHIFTTIME=1;
23     }
24     if( strcmp(argv[i],"-p")==0){
25         FUNC16=1;
26     }
27     if( strcmp(argv[i],"-cut")==0){
28         cut_frag=1;
29     }
30     if( strcmp(argv[i],"-Tsys")==0){
31         TSYS_FRAG=1;
32     }
33     if( strcmp(argv[i],"-pow")==0){
34         pow_frag=1;
35     }
36     if( strcmp(argv[i],"-check")==0){
37         check_frag=1;
38     }
39 }
40
41 if(check_frag==0){
42 if(pow_frag==0){
43 if(takahagi_frag==0){
44 h_func1();
45 h_func2();
46 h_func3();
47 h_func4();
48 h_func5();
49 h_func6();
50 h_func7();
51 h_func8();
52 h_func9();
53 h_func10();
54 h_func11();
55 h_func12();
56 h_func13();
57 h_func13();
58 h_func14();
59 h_func15();
60 if(TSYS_FRAG==1){
61 h_plus1_1();
62 }
63 if(cut_frag==1){
64 h_plus2_1();
65 }
66 h_func16();
67 }
68
69 else if(takahagi_frag==1){
70 t_func1();
71 t_func2();
72 t_func3();
73 t_func4();
74 t_func5();
75 t_func6();
76 t_func7();
77 t_func8();
78 t_func9();
79 t_func10();
80 t_func11();
81 t_func12();
82 t_func13();
83 t_func14();
84 t_func15();
85 if(TSYS_FRAG==1){
86 t_plus1_1();
87 }
88 if(cut_frag==1){
89 t_plus2_1();
90 }
91 t_func16();
92 }
93
94 if(TSYS_FRAG==1){
95 plus1_2();
96 }
97 }
98
99
100 else if(pow_frag==1){
101 if(TSYS_FRAG==0){
102 plus3_1();
103 }
104 else if(TSYS_FRAG==1){
105 plus3_2();
106 }
107 }
108 }
109
110
111 else if(check_frag==1){
112 plus4_1();
113 }
114
115
116 return 0;
117 }

```

h_func15.c


```

47 fclose(fp);
48 fclose(gp);
49 return 0;
50 }

```

plus3_2.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<math.h>
4  #include <string.h>
5  #include <sys/stat.h>
6  #include "head.h"
7
8  int plus3_2(void){
9
10 FILE*fp;
11 FILE*gp;
12
13 double p1,p2,p3,p2mW,p3mW,Tsys2star,Tsys3star,tem,Tamb,PambdBm,PambmW;
14 char buf[LINE_SIZE],input[FILE_SIZE],output1[FILE_SIZE],output2[FILE_SIZE],dir[FILE_SIZE];
15
16 printf(" Please enter input file name of power log > ");
17 scanf("%s",input);
18
19 printf(" Please enter the temperature [degrees Celsius] > ");
20 scanf("%lf",&tem);
21
22 printf(" Please enter the P(Tamb) [dBm] > ");
23 scanf("%lf",&PambdBm);
24
25 Tamb=tem+273.15;
26 PambmW=pow(10,PambdBm/10);
27
28 strcpy(dir,input);
29 strcat(dir,"_Tstar");
30
31 if(mkdir(dir,S_IRUSR | S_IWUSR | S_IXUSR |S_IRGRP | S_IWGRP | S_IXGRP |S_IROTH | S_IXOTH | S_IXOTH) ==0){
32     printf(" Create a directory for '%s'\n",dir);
33 }
34
35 strcpy(output1,dir);
36 strcat(output1,"/");
37 strcat(output1,input); // input_Tstar/input
38
39 strcpy(output2,dir);
40 strcat(output2,"/");
41 strcat(output2,"log"); // input_Tstar/log
42
43 if((fp=fopen(input,"r"))==NULL){
44     printf(" Can not read the input file\n");
45     exit(1);
46 }
47
48 if((gp=fopen(output1,"w"))==NULL){
49     printf(" Unable to open output file\n");
50     exit(1);
51 }
52
53 while( fgets(buf,LINE_SIZE,fp)!=NULL ){
54     sscanf(buf,"%lf%lf%lf",&p1,&p2,&p3);
55     p2mW=pow(10,p2/10);
56     p3mW=pow(10,p3/10);
57     Tsys2star=Tamb*p2mW/(PambmW-p2mW);
58     Tsys3star=Tamb*p3mW/(PambmW-p3mW);
59     fprintf(gp,"%3lf %4lf %10lf %4lf %4lf %10lf %4lf\n",p1,p2,p2mW,Tsys2star,p3,p3mW,Tsys3star);
60 }
61
62 fclose(fp);
63 fclose(gp);
64
65
66 if((fp=fopen(output2,"w"))==NULL){
67     printf(" Unable to open output file\n");
68     exit(1);
69 }
70
71 fprintf(fp,"Tamb[K] P(Tamb) [dBm] P(Tamb) [mW]\n");
72 fprintf(fp,"%2lf %4lf %10lf\n",Tamb,PambdBm,PambmW);
73
74 fclose(fp);
75
76 return 0;
77 }
78

```

plus4_1.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include <math.h>
4
5  int plus4_1(void){
6
7
8  double AD,t,year,day,hour,minute,second,Year,Day,Hour,Minute;
9
10
11 printf(" Please enter the time of angle log\n");
12 printf("> ");
13 scanf("%lf",&AD);
14
15 if(AD>1.9E+12 && AD<2.1E+12){
16     year=floor(AD/pow(10,9)); //2010
17     day=floor(AD/pow(10,6))-1000*year; //187
18     hour=floor(AD/pow(10,4))-pow(10,5)*year-100*day; //6
19     minute=floor(AD/100)-pow(10,7)*year-pow(10,4)*day-100*hour; //20
20     second=AD-pow(10,9)*year-pow(10,6)*day-pow(10,4)*hour-100*minute; //43
21     Day=(year-1)*365+floor((year-1)/4)-floor((year-1)/100)+floor((year-1)/400)+(day-1)+2; //総日数 (+2日)
22     Hour=24*Day+hour;

```

```
23     Minute=60*Hour+minute;
24     t=60*Minute+second;
25
26     printf(" %.01f\n",t);
27
28 }
29
30 else{
31     printf(" Please enter the appropriate number\n");
32     exit(1);
33 }
34
35 return 0;
36 }
```