

# 構造体の練習問題のプログラム例

山本昌志\*

2004 年 5 月 19 日

## 1 本日の学習内容

教科書 p.288 練習問題 8 のプログラム例を示し、その動作を詳細に説明する。これらのプログラムは、これまで学習してきたことの全てが含まれている。動作の分からないところがあれば、教科書などを見て、完全に理解する必要がある。

ここでのメインテーマは構造体である。構造体の基本的な概念は、「プログラマーが分かりやすいように、データをまとめて管理する」と言うことである。これが便利なものであると理解し、自由自在に使いこなせなくてはならない。

## 2 Lesson 8-1 アンケート集計

### 2.1 問題

Lesson 8-1

以下のような予想結果を 10 人分入力し、男女別の予想結果を「勝ち・引き分け・負け」に分けて集計し、表示するプログラムを構造体を利用して作成しなさい。

あるスポーツの勝敗予想

1 人目の予想 (男性:M)    ホームチーム    勝ち:1

2 人目の予想 (男性:F)    ホームチーム    負け:2

⋮

⋮

⋮

10 人目の予想 (男性:F)    ホームチーム    引き分け:0

\*独立行政法人 秋田工業高等専門学校 電気情報工学科

## 2.2 問題分析と構造体

### 2.2.1 問題分析

問題は単純である。10 人分の予想結果を入力し、男女別に勝敗の予想人数を集計し、表示するプログラムである。ここで構造体を使うとしたら、各人の情報 (データ) に対してである。各人が持つ情報は、

- 性別
- ホームチームの勝敗予想

の 2 種類である。ということで、図 1 のように 10 個のこれらを格納する構造体を使うことにする。

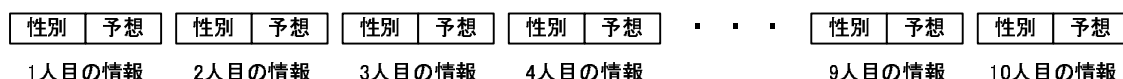


図 1: 勝敗予想の基本情報

集計結果も構造体にすることも可能であるが、教科書の解答の通り、配列を使うことにする。集計結果は全て男女別の勝敗予想の人数で、同じ性質の整数の集まりなので、配列でも十分である。

### 2.2.2 構造体の定義

各人の情報は性別と勝敗予想である。それが 10 人分ある。10 人同じ情報があるので、それは配列にするのが良いだろう。各個人の情報は、構造体で一つの変数にまとめると分かりやすくて良い。性別と勝敗予想を一つの変数にまとめるために、C 言語では次のように書く。

```
struct datatype{
    char sex;
    int toto;
};
```

datatype という構造体を定義し、その中には性別を表す文字型 (char) の変数 sex と、勝敗予想を表す整数型 (int) の変数 toto がある。これら 2 つの変数を構造体 datatype のメンバーと言う。これで、図 2 のような変数が定義できたのである。

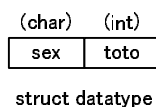


図 2: 構造体のイメージ。構造体 (struct datatype) は 2 つのメンバーがある。

文字型変数 sex に 1 文字を格納して各人の性別を表す。例えば、メンバーの sex は

- f が格納されていれば女性 (female)
- m が格納されていれば男性 (male)

とする。整数型変数 `toto` には、ホームチームの試合の予想結果を表す。例えば、メンバー `toto` は、ホームチームの勝敗として

- 0 が格納されていれば、引き分け
- 1 が格納されていれば、勝ち
- 2 が格納されていれば、負け

とする<sup>1</sup>。ここで、メンバー `sex` が f や m が男性や女性、0,1,2 が引き分けや勝ち負けを表すと決めたのはプログラマーであり、コンピューターはなにも関知していない事を忘れてはならない。

これで、データ構造の内容が決まった。新しい変数の型として構造体を定義した。まだ、データの型の定義が終わっただけで、コンピューターのメモリーの確保が終わっていない。実際にデータを操作する場合、メモリーを確保する事が必要である。これは、プログラム中で

```
struct datatype data[10];
```

と書けばよい。そうすると、図 3 のようにメモリーが確保される。

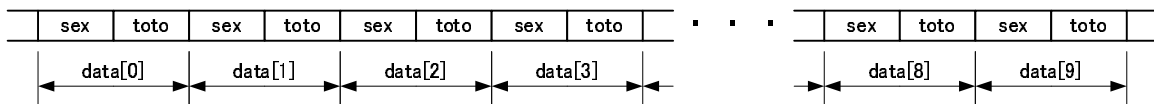


図 3: メモリー中の構造体のデータ

### 2.2.3 構造体のメンバーへのアクセス

これで、データの構造が決まったので、あとはそれぞれのメンバーにアクセスする方法が分かれば、構造体の全てが分かる。構造体のメンバーへのアクセスの前に通常の変数と配列のデータへのアクセスの方法を復習する。通常の変数、例えば `a` という変数へのアクセスは容易で、変数名を使えばそのデータを参照できる。次に配列であるが、それは配列名と添え字を示すことによりデータにアクセスできる。例えば、`b[3]` とかである。

データは構造体のメンバーに格納されているから、それにアクセスするためには、構造体名とメンバー名を示す必要がある。実際のアクセスの方法については、既に学習したように、ドット演算子をつかう。ここでは、次のように使われている。

```
data[loop].sex
data[loop].toto
```

<sup>1</sup>ここでも文字型の変数を使えるが、教科書の通り整数型を使う。

## 2.3 プログラム例

この問題のプログラム例 をリスト 1 に示す。このプログラムは、グローバルな部分と 4 つの関数からできておりそれぞれの動作は以下の通りである。

### 2.3.1 グローバル

10-13 行 構造体の定義

17-19 行 関数プロトタイプ宣言

### 2.3.2 main 関数

25 行 メイン関数の始まり。メイン関数の終わりは、39 行である。

28 行 構造体変数の宣言。これは、配列型の構造体を宣言して、メモリーを確保している。

29 行 集計結果を格納する配列の宣言。この配列には、以下のようなデータが格納される。

```
result[0][0] : 女性で引き分けと予想した人数
result[0][1] : 女性で勝ちと予想した人数
result[0][2] : 女性で負けと予想した人数
result[1][0] : 男性で引き分けと予想した人数
result[1][1] : 男性で勝ちと予想した人数
result[1][2] : 男性で負けと予想した人数
```

33 行 データ入力用の関数の呼び出し。

34 行 結果集計の関数の呼び出し。

35 行 結果出力の関数の呼び出し。

38 行 メイン関数が戻り値を返して、呼びだし元へ戻る。

39 行 メイン関数のブロックの終わりを示している。

### 2.3.3 データ入力関数 (data\_input)

55 行 データ入力関数の開始

56 行 ループのカウンターに用いる変数宣言

57 行 キーボード入力の時にバッファに残るゴミを入れる。

62-68 行 for 文による 10 回のループ

63 行 性別の入力を促す。

64 行 scanf 関数によりキーボードからデータを取得して、構造体変数 data[loop] のメンバー sex に格納する。

66 行 ホームチームの勝敗の入力を促す。

67 行 scanf 関数によりキーボードからデータを取得して、構造体変数 data[loop] のメンバー toto に格納する。

70 行 関数が戻り値を返して、呼びだし元へ戻る。

71 行 関数のブロックの終わりを示している。

### 2.3.4 データの集計関数 (calc\_result)

93 行 データ入力関数の開始

94 行 ループのカウンターに用いる変数宣言

97-101 行 人数を格納する配列をゼロに初期化する。

103-109 行 男女別勝敗の人数のカウント

111 行 関数が戻り値を返して、呼びだし元へ戻る。

112 行 関数のブロックの終わりを示している。

### 2.3.5 結果出力関数 (print\_result)

これは説明するまでもないであろう。各自考えよ。

### 2.3.6 プログラムリスト

リスト 1: アンケートの集計

```
1  /*=====*/
2  /*=====*/
3  /*      教科書「C言語プログラミング初歩の初歩」      */
4  /*      練習問題 8 Lesson 8-1 アンケートの集計      */
5  /*=====*/
6  /*=====*/
7  #include <stdio.h>
8
9  /* ---- 構造体 ---- */
10 struct datatype{
11     char sex;          /* 性別を表す文字      m:男性  f:女性      */
12     int  toto;         /* 予想結果を記憶する整数 0:引分 1:勝 2:負 */
13 };
14
15 /* ---- プロトタイプ宣言 ---- */
16
```

```

17 int data_input(struct datatype data[]);
18 int calc_result(struct datatype data[], int result[][3]);
19 int print_result(int result[][3]);
20
21
22 /*=====*/
23 /*          メイン 関数          */
24 /*=====*/
25 int main(void){
26
27     /* ---- 変数宣言 ---- */
28     struct datatype data[10]; /* 構造体の宣言 */
29     int result[2][3];         /* 勝敗予想人数[0:男 1:女][0:引分 1:勝 2:負]*/
30
31     /* ---- 実行文 ---- */
32
33     data_input(data);          /* データ入力関数呼び出し */
34     calc_result(data, result); /* 結果集計関数の呼び出し */
35     print_result(result);      /* 結果表示関数の呼び出し */
36
37
38     return 0;
39 }
40
41
42 /*=====*/
43 /* データの入力 */
44 /* [引数] */
45 /* 仮引数 data[] : 予想結果を格納する構造体 参照渡し */
46 /* [戻り値] */
47 /* int : 常に 0 が返される */
48 /* [動作内容] */
49 /* ユーザーに入力を促し、キーボードから入力された 男性 or 女性 と */
50 /* 勝敗予想を引数で渡された構造体に格納する。 */
51 /* 男性 f 女性 m */
52 /* 勝ち 1 引き分け 0 負け 2 */
53 /* これらの値は、予想結果を格納する構造体 data[] に格納される */
54 /*=====*/
55 int data_input(struct datatype data[]){
56     int loop; /* 読み込み番号のカウンター */
57     char temp; /* バッファに残ったゴミを入れる */
58
59
60     /* ---- 勝敗予想の入力 ---- */
61
62     for(loop = 0; loop < 10; loop++){
63         printf("%d人目の予想をする方の性別は? 女性:f 男性:m\n", loop+1);
64         scanf("%c%c", &data[loop].sex, &temp);
65
66         printf("ホームチームの試合? 勝ち:1 引き分け:0 負け:2\n");
67         scanf("%d%c", &data[loop].toto, &temp);
68     }
69
70     return 0;
71 }
72
73
74 /*=====*/
75 /* データの集計 */
76 /* [引数] */
77 /* 仮引数 data[] : 予想結果が格納されている構造体 参照渡し */
78 /* result[][3] : 集計結果の集計を格納する配列 参照渡し */

```

```

79 /*                      result[0:女性 1:男性][0:引分 1:勝 2:負] */
80 /*  [戻り値]                      */
81 /*      int : 常に 0 が返される      */
82 /*  [動作内容]                      */
83 /*      予想結果の集計をおこなう。各場合における人数は配列result[][] */
84 /*      に格納される。                      */
85 /*      勝敗予想を引数で渡された構造体に格納する。          */
86 /*      result[0][0] : 女性で引き分けと予想した人数          */
87 /*      result[0][1] : 女性で勝ちと予想した人数              */
88 /*      result[0][2] : 女性で負けと予想した人数              */
89 /*      result[1][0] : 男性で引き分けと予想した人数          */
90 /*      result[1][1] : 男性で勝ちと予想した人数              */
91 /*      result[1][2] : 男性で負けと予想した人数              */
92 /*=====*/
93 int calc_result(struct datatype data[], int result[][3]){
94     int loop1, loop2;
95
96     /* ---- 初期化 ---- */
97     for(loop1=0; loop1<2; loop1++){
98         for(loop2=0; loop2<3; loop2++){
99             result[loop1][loop2]=0; /* 勝敗予想人数の配列を全てゼロに */
100         }
101     }
102
103     for(loop1=0; loop1<10; loop1++){
104         if(data[loop1].sex == 'f'){
105             result[0][data[loop1].toto]++; /* 女性の場合 */
106         }else{
107             result[1][data[loop1].toto]++; /* 弾性の場合 */
108         }
109     }
110
111     return 0;
112 }
113
114 /*=====*/
115 /*      結果表示                      */
116 /*=====*/
117 int print_result(int result[][3]){
118
119     printf("\n集計結果表示####\n");
120     printf("女性の予想は、[勝ち:%2d][引き分け:%2d][負け:%2d]\n",
121         result[0][1], result[0][0], result[0][2]);
122     printf("男性の予想は、[勝ち:%2d][引き分け:%2d][負け:%2d]\n",
123         result[1][1], result[1][0], result[1][2]);
124
125     return 0;
126 }

```

#### 実行結果

```

##集計結果表示##
女性の予想は、[勝ち: 2][引き分け: 2][負け: 1]
男性の予想は、[勝ち: 1][引き分け: 2][負け: 2]

```

## 3 Lesson 8-2 データをまとめて管理する 1

### 3.1 問題

Lesson 8-2

音楽 CD 10 枚に入っている曲 (各 CD には 6 曲) を入力し、テキストファイルに記録するプログラムを作成しなさい。ただし、作成するテキストファイル (CDdata.txt) は次の通りとします。

```
CD1 title: Zokkon C GENGO
Artist: Hoge Maru
song1: Ikeike hensuu
song2: Hairetu BANZAI
.
.
.
song6: Saraba Kozoutai
CD2 Title: Love Love C
Artist: etc etc
```

プログラム中の CD データの取り扱いには、構造体を使うこと。

### 3.2 プログラム例

この問題のプログラム例 をリスト 2 に示す。

リスト 2: CD のデータをファイル出力

```
1 #include <stdio.h>
2
3 /* ---- 構造体 ---- */
4 struct cd_data{
5     char title[64]; /* タイトル名を入れる配列 */
6     char artist[64]; /* アーティスト名を入れる配列 */
7     char song[6][64]; /* 曲名をいれる配列 6曲まで入力可 */
8 };
9
10 /* ---- プロトタイプ宣言 ---- */
11 int data_input(struct cd_data CD[]);
12 int write_file(struct cd_data CD[]);
13
14 /*=====*/
15 /*          メイン関数          */
16 /*=====*/
17 int main(void){
18     struct cd_data CD[10];
19
20     data_input(CD); /* CDのデータを入れる関数呼び出し */
21     write_file(CD); /* CDのデータをファイル出力する関数呼び出し */
22
23     return 0;
24 }
```



```

25
26 /*=====*/
27 /*          データの入力          */
28 /*=====*/
29 int data_input(struct cd_data CD[]){
30     int i,j;
31
32     for(i=0; i<10; i++){
33
34         printf("\n-----\n");
35         printf("%d 番目のCDについてです。 \n", i+1);
36
37         printf("タイトル?\t");
38         fgets(CD[i].title,64,stdin);    /* 空白を含めてタイトル読み込み */
39
40         printf("\nアーティスト?\t");
41         fgets(CD[i].artist,64,stdin);    /* 空白を含めてアーティスト読み込み */
42
43         for(j=0; j<6; j++){
44             printf("\n\t%d 番目の曲目?\t", j+1);
45             fgets(CD[i].song[j],64,stdin); /* 空白を含めて曲名読み込み */
46         }
47     }
48 }
49
50 return 0;
51
52 }
53
54 /*=====*/
55 /*          データのfile出力          */
56 /*=====*/
57 int write_file(struct cd_data CD[]){
58     FILE *wfile;
59     int i,j;
60
61     wfile = fopen("CDdata.txt","w");    /* 書き込み用fileのオープン */
62
63
64     /* ---- 10個のCDタイトルの書き込み ---- */
65     for(i=0; i<10; i++){
66
67         fprintf(wfile, "CD%d Title: %s", i+1, CD[i].title); /* タイトル */
68         fprintf(wfile, "Artist: %s", CD[i].artist);        /* アーティスト*/
69
70         for(j=0; j<6; j++){                                /* 曲名 */
71             fprintf(wfile, "song%d: %s", j+1, CD[i].song[j]);
72         }
73     }
74
75 }
76
77 fclose(wfile);    /* 書き込み用のファイルのクローズ */
78
79 return 0;
80
81 }

```

## 4 Lesson 8-3 データをまとめて管理する2

### 4.1 問題

Lesson 8-3

Lesson 8-2 で作成したテキストファイルを読み込み、曲名を入力するとその曲が入っている CD の名前と何番目かを表示してくれるプログラムを作成しなさい。

### 4.2 プログラム例

この問題のプログラム例 をリスト 3 に示す。

リスト 3: 曲名検索

```
1 #include <stdio.h>
2 #include <string.h>
3
4 /*----- 構造体 -----*/
5 struct cd_data{
6     char title[64];           /* タイトルを入れる配列 */
7     char artist[64];          /* アーティストを入れる配列 */
8     char song[6][64];         /* 曲名を入れる配列(6曲分) */
9 };
10
11 /*----- プロトタイプ宣言 -----*/
12 int read_file(struct cd_data CD[]);
13 int read_title(int cdnum, FILE *rfile, struct cd_data CD[]);
14 int read_artist(int cdnum, FILE *rfile, struct cd_data CD[]);
15 int read_song(int cdnum, FILE *rfile, struct cd_data CD[]);
16 int find_song(char song[], struct cd_data CD[]);
17
18 /*=====
19 /*             メイン関数                                     */
20 /*=====
21 int main(void){
22     char song[64];             /* 検索する曲名を入れる配列 */
23     int nc, find;
24     struct cd_data CD[10];     /* CDのデータを入れる構造体 */
25
26     /*----- 検索する曲名の入力 -----*/
27
28     printf("探したい曲名?\t");
29     fgets(song, 64, stdin);     /* キーボードから曲名入力 */
30     nc=strlen(song);            /* 曲名の文字数+1(\n) */
31     song[nc-1]='\0';            /* 改行(\n)を文字終了(\0)に変更 */
32     printf("\n");               /* 改行 */
33
34
35     /*----- ファイルの読み込みと検索 -----*/
36     read_file(CD);              /* CDのデータの読み込み */
37     find = find_song(song,CD);  /* 曲名の検索 */
38
39     /*----- 見つからなかった場合のメッセージ -----*/
40     if(find==0){
41         printf("お探しの曲はありません。 \n");
42     }
```

```

43     return 0;
44 }
45
46
47 /*=====*/
48 /*          データのfileの読み込み          */
49 /*=====*/
50 int read_file(struct cd_data CD[]){
51     FILE *rfile;
52     int i;
53
54     rfile = fopen("CDdata.txt","r");    /* CDのデータファイルのオープン */
55
56     for(i=0; i<10; i++){
57         read_title(i, rfile, CD);        /* タイトル名を読み込む関数のコール */
58         read_artist(i, rfile, CD);        /* アーティスト名を読み込む関数のコール */
59         read_song(i, rfile, CD);         /* 曲名を読み込む関数のコール */
60     }
61
62     fclose(rfile);                      /* ファイルのクローズ */
63
64     return 0;
65 }
66
67
68 /*=====*/
69 /*          タイトル名の読み込み          */
70 /*=====*/
71 int read_title(int cdnum, FILE *rfile, struct cd_data CD[]){
72     int nc, nc1, nc2;
73     char temp, c1[64], c2[64];
74
75     sprintf(c1,"CD%d",cdnum+1);        /* CD番号 CD1, CD2, CD3 ...をc1へ */
76     nc1=strlen(c1);                    /* CD番号の文字数 */
77
78     do{                                /* CD番号の検索 */
79         fscanf(rfile, "%s",c2);
80         nc2=strlen(c2);
81         if(nc1<nc2){
82             nc=nc2;
83         }else{
84             nc=nc1;
85         }
86     }while(strncmp(c1,c2,nc)!=0);
87
88     temp=fgetc(rfile);                 /* 空白の読み飛ばし */
89
90     sprintf(c1,"Title");
91     nc=strlen(c1);
92
93     do{                                /* Title: の検索 */
94         fscanf(rfile, "%s",c2);
95     }while(strncmp(c1,c2,nc)!=0);
96
97     temp=fgetc(rfile);                 /* 空白の読み飛ばし */
98
99     fgets(CD[cdnum].title,64,rfile); /* タイトルの読み込み */
100
101     nc=strlen(CD[cdnum].title);
102     CD[cdnum].title[nc-1]='\0';        /* 改行(\n)を文字終了(\0)に変更 */
103
104     return 0;

```

```

105 }
106 }
107
108 /*=====*/
109 /*          アーティスト名の読み込み          */
110 /*=====*/
111 int read_artist(int cdnum, FILE *rfile, struct cd_data CD[]){
112     int nc, nc1, nc2;
113     char temp, c1[64], c2[64];
114
115     sprintf(c1,"Artist:");          /* "Artist:"をc1へ格納 */
116     nc1=strlen(c1);
117
118     do{                                /* "Artist:"の検索 */
119         fscanf(rfile, "%s",c2);
120         nc2=strlen(c2);
121         if(nc1<nc2){
122             nc=nc2;
123         }else{
124             nc=nc1;
125         }
126     }while(strncmp(c1,c2,nc)!=0);
127
128     temp=fgetc(rfile);                /* 空白の読み飛ばし */
129
130     fgets(CD[cdnum].artist,64,rfile); /* 空白付きアーティストの読み込み */
131
132     nc=strlen(CD[cdnum].artist);
133     CD[cdnum].artist[nc-1]='\0';      /* 改行(\n)を文字終了(\0)に変更 */
134
135     return 0;
136 }
137 }
138
139 /*=====*/
140 /*          曲名の読み込み          */
141 /*=====*/
142 int read_song(int cdnum, FILE *rfile, struct cd_data CD[]){
143     int i, nc, nc1, nc2;
144     char temp, c1[64], c2[64];
145
146     for(i=0; i<6; i++){                /* 6曲分ファイルから読み込む */
147         sprintf(c1,"song%d:", i+1);    /* "song?:"をc1へ格納 */
148         nc1=strlen(c1);
149
150         do{                                /* "song?:"の検索 */
151             fscanf(rfile, "%s",c2);
152             nc2=strlen(c2);
153             if(nc1<nc2){
154                 nc=nc2;
155             }else{
156                 nc=nc1;
157             }
158         }while(strncmp(c1,c2,nc)!=0);
159
160         temp=fgetc(rfile);              /* 空白の読み飛ばし */
161
162         fgets(CD[cdnum].song[i],64,rfile); /* 曲名の読み込み */
163
164         nc=strlen(CD[cdnum].song[i]);
165         CD[cdnum].song[i][nc-1]='\0';    /* 改行(\n)を文字終了(\0)に変更 */
166

```

```

167     }
168
169     return 0;
170
171 }
172
173 /*=====*/
174 /*          曲名検索          */
175 /*=====*/
176 int find_song(char song[], struct cd_data CD[]){
177     int i,j;
178     int nc, nc1, nc2, fnd;
179
180     fnd=0;
181
182     nc1=strlen(song);
183
184     for(i=0; i<10; i++){          /* CD番号のループ */
185         for(j=0; j<6; j++){        /* 曲番号のループ */
186
187             nc2=strlen(CD[i].song[j]); /* 曲名の文字数 */
188             if(nc1<nc2){              /* 文字数の多い方で比較 */
189                 nc=nc2;
190             }else{
191                 nc=nc1;
192             }
193
194             /* ---- 曲名が一致したら画面へ出力 ---- */
195
196             if(strncmp(song, CD[i].song[j], nc)==0){
197                 if(fnd==0){
198                     printf("%sが見つかりました。タイトルと曲番は以下の通り。\\n", song);
199                     fnd=1;
200                 }
201                 printf("\\t%s    %d\\n", CD[i].title, j+1);
202             }
203         }
204     }
205 }
206
207 return fnd;          /* 0; 見つからない場合    1: 見つかった場合 */
208
209 }
210

```

### 実行結果

探したい曲名? song1

song1が見つかりました。タイトルと曲番は以下の通り。

```

title5    1
title6    1

```

## 5 レポート

### 5.1 内容

練習問題のうち、「Lesson 8-1 アンケート集計」と「Lesson 8-2 データをまとめて管理する 1」のプログラムを完成させること。実行の確認ができたならば、それをプリントアウトして、レポートとして提出する。さらに、使われている構造体のデータの内容を説明すること。

自宅のパソコンでプログラムを作っても良い。そして、実行の確認がとれたならば、それをプリントアウトし、レポートとして提出しても良い。

### 5.2 レポート 提出要領

提出方法は、次の通りとする。

期限	6月2日(木)PM5:00 まで
用紙	A4
提出場所	山本研究室の入口のポスト
表紙	表紙を1枚つけて、以下の項目を分かりやすく記述すること。 授業科目名「情報工学概論」 課題名「課題1 構造体」 2E 学籍番号 氏名 提出日
内容	ソースプログラム(プリントアウトのみ、手書きは不可) 使われている構造体の説明