

夏休みの課題のプログラム例

山本昌志*

2005 年 9 月 7 日

1 前期末試験の内容と本日の講義

前期末試験では、1 年生から学習してきた内容、全てにわたって試験を行う。試験で出題するプログラムは、夏休みの課題が中心である。夏休みの課題のプログラムが全て、自力で書けるように学習すること。

本日は夏休みの課題のプログラム全てを説明する。

2 課題プログラムの解答

2.1 基本プログラム

[練習 1] 以下の 1 文を表示する。

```
Hello World !!
```

リスト 1: 1 行表示

```
1 #include <stdio.h>
2
3 int main(){
4     printf("Hello World !!\n");
5
6     return 0;
7 }
8
```

[練習 2] 以下の 2 行を表示する。

```
Hello World !!
from Akita National College of Technology.
```

*独立行政法人 秋田工業高等専門学校 電気情報工学科

リスト 2: 2 行表示

```

1 #include <stdio.h>
2
3 int main(){
4
5     printf("Hello World !!\n");
6     printf("from Akita National College of Technology.\n");
7
8     return 0;
9 }

```

[練習 3] 2 つの整数をキーボードから読み込んで、和と差、積を表示する。

リスト 3: 整数の和と差、積

```

1 #include <stdio.h>
2
3 int main(){
4
5     int i,j;
6
7     /* —— 入力のメッセージと整数入力—— */
8     printf("最初の整数?\t");
9     scanf("%d",&i);
10    printf("次の整数?\t");
11    scanf("%d",&j);
12
13    /* —— 計算と表示 —— */
14    printf("%d+%d=%d\n",i,j,i+j);
15    printf("%d-%d=%d\n",i,j,i-j);
16    printf("%d*%d=%d\n",i,j,i*j);
17
18    return 0;
19 }

```

[練習 4] 2 つの整数をキーボードから読み込んで、商と余りを表示する。

リスト 4: 整数の商と余り

```

1 #include <stdio.h>
2
3 int main(){
4
5     int i,j;
6
7     /* —— 入力のメッセージと整数入力—— */
8     printf("最初の整数?\t");
9     scanf("%d",&i);
10    printf("次の整数?\t");
11    scanf("%d",&j);
12
13
14    /* —— 計算と表示 —— */
15    printf("%d/%d=\n",i,j);
16    printf("\t商\t%d\n",i/j);
17    printf("\t余り\t%d\n",i%j);
18
19    return 0;
20 }

```

[練習 5] 2つの実数をキーボードから読み込んで、和と差、積、商を表示する。

リスト 5: 実数の四則演算

```
1 #include <stdio.h>
2
3 int main(){
4
5     double x, y;
6
7     /* —— 入力のメッセージと整数入力—— */
8     printf("最初の実数?\t");
9     scanf("%lf",&x);
10    printf("次の実数?\t");
11    scanf("%lf",&y);
12
13    /* —— 計算と表示 —— */
14    printf("%f+%f=%f\n",x,y,x+y);
15    printf("%f-%f=%f\n",x,y,x-y);
16    printf("%f*%f=%f\n",x,y,x*y);
17    printf("%f/%f=%f\n",x,y,x/y);
18
19    return 0;
20 }
```

[練習 6] 角度 [deg] を読み込んで、それをラジアンに直して、表示する。

リスト 6: 度からラジアンへの変換

```
1 #include <stdio.h>
2
3 int main(){
4
5     double rad, deg;
6
7     /* —— 入力のメッセージと整数入力—— */
8     printf("角度 [deg]?\t");
9     scanf("%lf",&deg);
10
11    /* —— 計算と表示 —— */
12    rad=deg/180.0*3.14159265;
13    printf("%f [rad]\n",rad);
14
15    return 0;
16 }
```

[練習 7] 角度 [deg] を読み込んで、角度と三角関数の値を表示する。

リスト 7: 三角関数の値の表示

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(){
5
6     double rad, deg;
7
8     /* —— 入力のメッセージと整数入力—— */
9     printf("角度 [deg]?\t");
```

```

10     scanf("%lf",&deg);
11
12     /* —— 計算と表示 —— */
13     rad=deg/180.0*3.14159265;
14     printf("sin = %f\n",sin(rad));
15     printf("cos = %f\n",cos(rad));
16     printf("tan = %f\n",tan(rad));
17
18     return 0;
19 }

```

2.2 制御文

[練習 1] if と goto 文を使って、1~1000 までの整数の合計を求める。

リスト 8: twif と goto を使ったループ

```

1  #include <stdio.h>
2
3  int main(){
4
5      int i, sum;
6
7      /*—— 初期化 ——*/
8      sum=0;
9      i=1;
10
11     loop:                                /* ラベル */
12     sum+=i;                               /* sum=sum+i */
13     i++;                                  /* iのインクリメント */
14     if(i<=1000)goto loop;
15
16     printf("sum = %d\n",sum);
17
18     return 0;
19 }

```

[練習 2] for 文を使って、1~1000 までの整数の合計を求める。

リスト 9: for を使ったループ

```

1  #include <stdio.h>
2
3  int main(){
4
5      int i, sum;
6
7      /*—— 初期化 ——*/
8      sum=0;
9
10     for(i=1; i<=1000; i++){
11         sum+=i;                               /* sum=sum+i */
12     }
13
14     printf("sum = %d\n",sum);
15
16     return 0;

```

```
17 }
```

[練習 3] while 文を使って、1～1000 までの整数の合計を求める。

リスト 10: while を使ったループ

```
1 #include <stdio.h>
2
3 int main(){
4
5     int i, sum;
6
7     /*—— 初期化 ——*/
8     sum=0;
9     i=1;
10
11     while(i<=1000){
12         sum+=i;                /* sum=sum+i */
13         i++;
14     }
15
16     printf("sum = %d\n",sum);
17
18     return 0;
19 }
```

[練習 4] do～while 文を使って、1～1000 までの整数の合計を求める。

リスト 11: do～while を使ったループ

```
1 #include <stdio.h>
2
3 int main(){
4
5     int i, sum;
6
7     /*—— 初期化 ——*/
8     sum=0;
9     i=1;
10
11     do{
12         sum+=i;                /* sum=sum+i */
13         i++;
14     }while(i<=1000);
15
16     printf("sum = %d\n",sum);
17
18     return 0;
19 }
```

[練習 5] 1～100000 までの素数を全て書き出す

リスト 12: 繰り返し文を用いた素数の抽出

```
1 #include <stdio.h>
2 #include <math.h>
3 #define TEST 100000
```

```

4
5 int main(void){
6     int number[TEST+1];
7     int i,j,test_max;
8
9     /*—— 初期化 ——*/
10    for(i=1;i<=TEST;i++){          /* number[i]=0    iは素数 */
11        number[i]=0;                /* number[i]=1    iは素数でない */
12    }
13
14    /*—— 素数判定(エラトステネスのふるい) ——*/
15    test_max=(int)sqrt(TEST);
16    for(i=2;i<=test_max;i++){
17        if(number[i]==0){
18            for(j=2*i; j<=TEST; j+=i){
19                number[j]=1;
20            }
21        }
22    }
23
24    /*—— 結果表示 ——*/
25    j=0;
26    printf("No\tPrime Number\n");
27
28    for(i=2;i<=TEST;i++){
29        if(number[i]==0){
30            j++;
31            printf("%d\t%d\n",j,i);
32        }
33    }
34
35
36    return 0;
37 }

```

2.3 配列

[練習 1] キーボードから 10 個の整数の値を読み込んで、それを配列に格納する。偶数の合計と奇数の合計を計算するプログラムを作成せよ。

リスト 13: 配列の利用

```

1 #include <stdio.h>
2 #define N 10
3
4 int main(){
5     char temp;
6     int i, input_num[N];
7     int sum_gusu, sum_kisu;
8
9     /* —— キーボードから読み込んだ整数を配列へ格納 —— */
10    for(i=0; i<N; i++){
11        printf("%d 番目の整数? \t", i+1);
12        scanf("%d%c", &input_num[i], &temp);
13    }
14
15    /* —— 合計を入れる変数を 0 に初期化 —— */
16    sum_gusu = 0;

```

```

17     sum_kisu = 0;
18
19     /* —— 偶数と奇数の合計の計算 —— */
20     for (i=0; i<N; i++){
21         if (input_num[i]%2 == 0){           /* 偶数ならば */
22             sum_gusu += input_num[i];
23         } else {                             /* さもなければ */
24             sum_kisu += input_num[i];
25         }
26     }
27
28
29     printf("偶数の和\t%d\n", sum_gusu);
30     printf("奇数の和\t%d\n", sum_kisu);
31
32     return 0;
33 }

```

[練習 2] 数字を大きい、あるいは小さい順に並べることをソーティングと言う。最も簡単な単純挿入法のプログラムを作成する。

リスト 14: 単純挿入法

```

1  #include <stdio.h>
2  #include <stdlib.h>      /* 乱数発生のため */
3  #include <time.h>        /* 時刻の関数を使うため */
4
5  int main(void){
6      int a[1024], i, j, ndata, test;
7
8      ndata=1024;
9
10     srand((unsigned int)time(NULL)); /* 起動毎に異なる乱数を発生させるため */
11
12     for (i=0; i<ndata; i++){
13         a[i]=rand();                /* 配列 a[i] に乱数の整数を設定 */
14     }
15
16
17     j=1;
18
19     /* —— 単純ソート —— */
20     while(j<ndata){
21         test = a[j];
22         i=j-1;
23
24         while(0<=i && test<a[i]){    /* 大きい方から比較 */
25             a[i+1] = a[i];
26             i--;
27         }
28
29         a[i+1] = test;
30         j++;
31     }
32
33     /* —— 昇順に並んだ結果の表示 —— */
34     for (i=0; i<ndata; i++){
35         printf("%d\n", a[i]);
36     }
37 }

```

```

38     return 0;
39 }

```

2.4 ファイル入出力

[練習 1] ファイルに、0 ~ 360 度の三角関数の値を書き込むプログラムを作成する。ファイルの各行には、角度 θ [deg] と $\sin \theta$ 、 $\cos \theta$ 、 $\tan \theta$ の値をタブ区切りで書くものとする。

リスト 15: 三角関数表の作成

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5
6      int theta;
7      double rad, s, c, t;
8      FILE *fp;
9
10     fp=fopen("triangle_table.txt","w"); /* ファイルのオープン */
11
12     for(theta=0; theta<=360; theta++){
13         rad=theta/180.0*3.1415926; /* 度からラジアンへ */
14
15         s=sin(rad); /* 三角関数の計算 */
16         c=cos(rad);
17         t=tan(rad);
18
19         fprintf(fp,"%d\t%f\t%f\t%f\n", theta, s, c, t); /* 書き込み */
20     }
21
22     fclose(fp); /* ファイルのクローズ */
23
24     return 0;
25 }

```

[練習 2] 前問で作成したファイルを読み込み、ディスプレイに表示するプログラムを作成せよ。

リスト 16: 三角関数表の読み込みと表示

```

1  #include <stdio.h>
2
3  int main(){
4
5      int theta;
6      double s, c, t;
7      FILE *fp;
8
9      /* —— ファイルのオープン(読み込みモード) —— */
10     fp=fopen("triangle_table.txt","r");
11
12     /* —— ファイルの読み込みと表示 —— */
13     while(fscanf(fp, "%d%lf%lf%lf", &theta, &s, &c, &t) != EOF){
14         printf("%d\t%f\t%f\t%f\n", theta, s, c, t); /* 表示 */
15     }
16 }

```



```

17     fclose(fp);                                /* ファイルのクローズ */
18
19     return 0;
20 }

```

2.5 文字処理

[練習 1] キーボードから 32 文字以内のアルファベットを入力する。入力した順序と逆にディスプレイに表示するプログラムを作成せよ。

リスト 17: 読み込んだ文字を逆に表示

```

1  #include <stdio.h>
2  #include <string.h>
3
4  int main(){
5
6      char moji[32];
7      int i, n;
8
9      printf("32文字以内のアルファベットを入力してください\n");
10     fgets(moji, 32, stdin);      /* 空白を含め1行読み込み */
11     n=strlen(moji);              /* 読み込んだ文字数カウント */
12
13
14     /* —— 文字が格納されている配列を逆から表示 —— */
15     for(i=n-2; i>=0; i--){
16         printf("%c", moji[i]);
17     }
18
19     printf("\n");
20
21     return 0;
22 }

```

2.6 関数

[練習 1] 三辺の長さ (a, b, c) をキーボードから入力して、面積を求めるプログラムを作成する。

リスト 18: ヘロンの公式により三角形の面積を求める

```

1  #include <stdio.h>
2  #include <math.h>
3
4  double heron(double x, double y, double z);
5  /* ===== */
6  /*      main 関数      */
7  /* ===== */
8  int main(){
9
10     double a, b, c, s;
11     char temp;
12
13     printf("三角形の3辺の長さを入力してください\n");

```

```

14     scanf("%lf%lf%lf%c", &a, &b, &c, &temp);
15
16     s=heron(a, b, c);
17
18     if(s<-900){
19         printf("入力した3辺では三角形は出来ません\n");
20     }else{
21         printf("面積は、%fです。 \n", s);
22     }
23
24     return 0;
25 }
26
27
28 /* ===== */
29 /*      ヘロンの公式より、三角形の面積を求める関数      */
30 /* ===== */
31 double heron(double x, double y, double z){
32     double s, menseki;
33
34     if((x+y)<z || (y+z)<x || (z+x)<y){
35         return -999;
36     }
37
38     s=(x+y+z)/2.0;
39
40     menseki=sqrt(s*(s-x)*(s-y)*(s-z));
41
42     return menseki;
43 }

```

2.7 構造体

- [練習 1] 以下のような 50 人分の成績のファイルがある。ファイルの各行には、名字、名前、英語、数学、情報処理の成績が書かれている。これらを構造体を使って管理し、平均点の高い順に学生の情報をディスプレイに表示する。表示する情報は、順位、平均点、名字、名前、英語、数学、情報処理の成績とする。

リスト 19: 成績を表示する

```

1  #include <stdio.h>
2  #define NINZU 45
3
4  struct kanri{
5      char myoji[32];
6      char namae[32];
7      int jyuni;
8      double heikin;
9      int english;
10     int math;
11     int info;
12 };
13
14 int read(struct kanri g[]);
15 int sekiji_keisan(struct kanri g[]);
16 int hyoji(struct kanri g[]);
17

```

```

18  /* ===== */
19  /*      main 関数      */
20  /* ===== */
21  int main(){
22      struct kanri gakusei[NINZU];
23
24      read(gakusei);
25      sekiji_keisan(gakusei);
26      hyoji(gakusei);
27
28      return 0;
29  }
30
31
32  /* ===== */
33  /*      データを読み込む関数      */
34  /* ===== */
35  int read(struct kanri g[]){
36      FILE *data;
37      int i;
38
39      data=fopen("seiseki.txt","r");
40
41      for(i=0;i<NINZU;i++){
42          fscanf(data,"%s%s%d%d", g[i].myoji, g[i].namae,
43                  &g[i].english, &g[i].math, &g[i].info);
44      }
45
46      fclose(data);
47
48      return 0;
49  }
50
51
52  /* ===== */
53  /*      平均を計算して、席次を決める関数      */
54  /* ===== */
55  int sekiji_keisan(struct kanri g[]){
56      int i, j;
57      struct kanri test;
58
59      for(i=0;i<NINZU;i++){
60          g[i].heikin=(g[i].english+g[i].math+g[i].info)/3.0;
61      }
62
63      /* —— 単純挿入法(降順) —— */
64      j=1;
65      while(j<NINZU){
66          test = g[j];
67          i=j-1;
68
69          while(0<=i && test.heikin>g[i].heikin){
70              g[i+1] = g[i];
71              i--;
72          }
73
74          g[i+1] = test;
75          j++;
76      }
77
78      return 0;
79  }

```

```

80
81
82 /* ===== */
83 /*      成績順に表示する関数      */
84 /* ===== */
85 int hyoji(struct kanri g[]){
86     int i;
87
88     for(i=0;i<NINZU;i++){
89         printf("%d\t%lf\t%s\t%s\t%d\t%d\t%d\n",
90             i+1,
91             g[i].heikin,
92             g[i].myoji,
93             g[i].namae,
94             g[i].english,
95             g[i].math,
96             g[i].info
97         );
98     }
99
100     return 0;
101 }

```

2.8 ポインター

〔練習 1〕 整数型の変数、a, b, c に入っている整数を、 $a \rightarrow b$ 、 $b \rightarrow c$ 、 $c \rightarrow a$ と入れ替えるプログラムを作成せよ。

リスト 20: 変数の値をサイクリックに入れ替える

```

1 #include <stdio.h>
2
3 int rot(int *x, int *y, int *z);
4 /* ===== */
5 /*      main 関数      */
6 /* ===== */
7 int main(){
8
9     int a, b, c;
10
11     a=1;
12     b=2;
13     c=3;
14
15     printf("a=%d\tb=%d\tc=%d\n", a, b, c);
16
17     rot(&a, &b, &c);    /* 入れ替えの関数 */
18
19     printf("a=%d\tb=%d\tc=%d\n", a, b, c);
20
21
22     return 0;
23 }
24
25
26 /* ===== */
27 /*      値をサイクリックに入れ替える      */
28 /* ===== */

```

```

29 int rot(int *x, int *y, int *z){
30     int temp;
31
32     temp=*z;
33
34     *z = *y;
35     *y = *x;
36     *x = temp;
37
38     return 0;
39 }

```

[練習 2] 要素数が 100×100 の 2 次元配列に $a[i][j]=i*i+j$ の値を代入する。そして、 $a[i][j]$ と $a[j][i]$ と入れ替えるプログラムを作成せよ。

リスト 21: 転置行列

```

1  #include <stdio.h>
2  #define N 100
3
4  int tranpose(int matrix[][N]);
5  /* ===== */
6  /*      main 関数      */
7  /* ===== */
8  int main(){
9
10     int i, j, matrix[N][N];
11
12     for(i=0; i<N; i++){
13         for(j=0; j<N; j++){
14             matrix[i][j] = i*i+j;
15         }
16     }
17
18     tranpose(matrix);
19
20     return 0;
21 }
22
23
24 /* ===== */
25 /*      値をサイクリックに入れ替える      */
26 /* ===== */
27 int tranpose(int a[][N]){
28     int i, j, temp;
29
30     for(i=0; i<N; i++){
31         for(j=i+1; j<N; j++){
32             temp=a[i][j];
33             a[i][j]=a[j][i];
34             a[j][i]=temp;
35         }
36     }
37
38     return 0;
39 }

```