

# プログラミング基礎 第12回

藤江 真也  
2021年7月9日

## スケジュール

- 7月8日(木) 23:59
  - 課題再提出期限  
(詳細は近日中にmanabaで公開)
- 7月9日(金) 第12回
  - 課題解答例, 練習問題公開
  - 最終課題作成
- 7月13日(火) 23:59
  - 最終課題提出期限
- 7月16日(金) 第13回
  - 最終課題 講評, 練習問題 解答と解説
  - 期末試験

## 7月16日(金) 第13回の詳細

- 最終課題講評(12号館3階)
  - 9:00~10:00  
Aグループ(21C1001~21C1073)
  - 10:00~11:00  
Bグループ(21C1074~21C1141, 再履修等)
- 期末試験 11:30~12:30
  - 8号館8101講義室  
Aグループ(21C1001~21C1073)
  - 8号館8208講義室  
Bグループ(21C1074~21C1141, 再履修等)

## 本日の内容

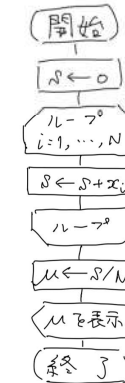
- 課題の解答例の公開(この資料)
- 期末試験練習問題
- 最終課題の作成

## 課題 解答例

## 第2回

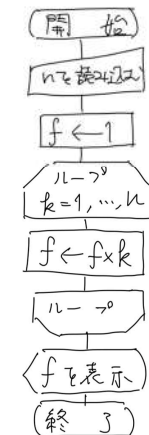
### ■ 課題 2.1

- $N$ 個の数値  $x_1, x_2, x_3, \dots, x_N$  の平均値  $\mu$  を求めて画面に表示するアルゴリズムのフローチャートを作図せよ



### ■ 課題 2.2

- 整数値を1つキーボードから読み込み、その階乗を求めて、画面に表示するアルゴリズムのフローチャートを作図せよ



## 第3回, 第4回

- 第3回(drawlibで自由な描画)は省略
- 第4回(wordで作成)は別途配布

## 第5回

### 課題

#### ■ 課題1

- コンピュータの中の実数(浮動小数点型)は、誤差を含む。それを実感するために、プログラムで求めた $\sqrt{2}$ の平方がぴったり2にならないことを確かめてください。
- kadai1.c 中の指示に従いプログラムを完成させて提出してください。
- 提出するファイルのファイル名は、20C1987\_kadai1.c のように、学籍番号\_kadai1.c にしてください。

```

1 /* 第5回 課題1 */
2 #include <stdio.h>
3 #include <math.h>
4
5 int main (void) {
6     float x, xsqrt;
7
8     printf("input X: ");
9     scanf("%f", &x);
10
11     /* %.2f の .2 は小数点以下2桁まで表示するという意味です */
12     printf("X = %.2f\n", x);
13     /* xsqrtにxの平方根を代入します */
14     xsqrt = sqrt(x);
15
16     /* x と xsqrt の2乗が等しいかどうか判定します */
17     if (x == xsqrt * xsqrt) {
18         printf("等しい\n");
19     } else {
20         printf("等しくない\n");
21     }
22
23     /* 小数点以下20ケタまで表示して、誤差を確認します */
24     printf("\n確認\n");
25     printf("X = %.20f\n", x);
26     printf("Xsqrt * Xsqrt = %.20f\n", xsqrt * xsqrt);
27
28     return 0;
29 }

```

## 実行結果例

```

fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai1
input X: 2
X = 2.00
等しくない

確認
X = 2.00000000000000000000
Xsqrt * Xsqrt = 1.99999988079071044922
fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai1
input X: 8
X = 8.00
等しくない

確認
X = 8.00000000000000000000
Xsqrt * Xsqrt = 7.99999952316284179688
fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai1
input X: 16
X = 16.00
等しい

確認
X = 16.00000000000000000000
Xsqrt * Xsqrt = 16.00000000000000000000

```

計算結果に誤差が含まれていることがわかる

## 実行結果例

```

fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai1
input X: 8.2
X = 8.20
等しい

確認
X = 8.19999980926513671875
Xsqrt * Xsqrt = 8.19999980926513671875
fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai1
input X: 8.3
X = 8.30
等しい

確認
X = 8.30000019073486328125
Xsqrt * Xsqrt = 8.30000019073486328125

```

小数点以下20桁まで表示させると、  
8.2や8.3などの小数は、計算する前から誤差を含んでいることがわかる

# 第5回

## 課題

### ■ 課題2

- 簡単な一次方程式のクイズを出すプログラム作成してください。
- kadai2.c の中の指示に従いプログラムを完成させて提出してください。
- 提出するファイルのファイル名は、  
20C1987\_kadai2.c  
のように、学籍番号\_kadai1.c にしてください。

```

1 /* 第5回 課題2 */
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <time.h>
5
6 int main(void) {
7     int xa, a, b, x;
8
9     srand((unsigned int)time(NULL));
10    /* rand() は0以上RAND_MAX (32767以上) 以下のランダムな整数を
11       戻り値とする関数です。整数Aで割ったあまりが0からA-1になる
12       ことを利用して、適当な範囲の乱数を生成できます。
13       適当な乱数Rをに対して整数AとBで R % A + B を計算することで、
14       BからA+B-1の範囲の乱数になります。 */
15    xa = rand() % 100; /* 0~99 */
16    a = rand() % 28 + 3; /* 3~30 */
17    b = rand() % 30 - 15; /* -15~14 */
18
19    printf("x * %d + %d = %d のときのxを答えよ\n", a, b, xa * a + b);
20    scanf("%d", &x);
21    if (x == xa) {
22        printf("正解\n");
23    } else {
24        printf("不正解\n");
25    }
26
27    return 0;
28 }

```

## 実行結果例

```

fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai2
x * 23 + -11 = 1047 のときのxを答えよ
46
正解
fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai2
x * 26 + 7 = 1489 のときのxを答えよ
56
不正解

```

bが負のときの見た目が悪いですが、これを直そうと思うと少々面倒です  
(bの値によって書式文字列を変更する必要がある)

## 改善例

```

19 if(b > 0) {
20     printf("x * %d + %d = %d のときのxを答えよ\n", a, b, xa * a + b);
21 } else if(b == 0) {
22     printf("x * %d = %d のときのxを答えよ\n", a, xa * a);
23 } else {
24     printf("x * %d - %d = %d のときのxを答えよ\n", a, -b, xa * a + b);
25 }

```

## 実行結果例

```

fujie@DESKTOP-KVRJVM:~/work/kadai_example/05$ ./kadai2r1
x * 22 - 6 = 1138 のときのxを答えよ
52
正解

```

# 第6回

## 課題

- 課題1 maxin.c を変更し、任意の個数の整数を標準入力から読み込み、その最大値、最小値、平均値(少数点以下3ケタ)を標準出力に出力するプログラムを作成せよ。

- ヒント: ファイルの最後まで読み込みが終わった場合、次の scanf 関数の戻り値は -1 になる。つまり、例えばint型の変数rを用意して、  
`r = scanf("%d", &x);`  
 とした場合、xへの数値の読み込みができなければrの値は -1 になる。

提出ファイルは「学籍番号\_kadai1.c」という名前にすること(例: 20C1981\_kadai1.c)

```

1 /* 第6回 課題1 */
2 #include <stdio.h>
3
4 int main(void) {
5     /* rはscanfの戻り値を受け取るための変数、
6     xはscanfで入力された値を受け取るための変数、 */
7     int r, x;
8     /* それぞれ、最大値、最小値、合計値、入力された回数を表す変数、
9     宣言されただけでは初期値が不定であることに注意 */
10    int max, min, sum, n;
11    /* 平均値は少数を含む可能性があるためfloat型にする */
12    float avg;
13
14    /* はじめの1回は、while文で繰り返す前に読み込み、
15    無条件で最大値、最小値に代入する。
16    ただし、この読み込みもできない可能性があることに注意 */
17    r = scanf("%d", &x);
18    if(r >= 0) {
19        max = x;
20        min = x;
21        sum = x;
22        n = 1;
23
24        /* 2回目以降はwhile文で繰り返す */
25        while(1) {
26            /* 値を読み込む。読み込めなかった場合はループを抜ける */
27            r = scanf("%d", &x);
28            if(r < 0)
29                break;

```

```

30        /* 読み込んだ値が最大値よりも大きければ、最大値を更新する */
31        if(x > max)
32            max = x;
33        /* 読み込んだ値が最小値よりも小さければ、最小値を更新する */
34        if(x < min)
35            min = x;
36        /* 合計値と読み込み回数を更新する */
37        sum += x;
38        n++;
39    }
40
41    /* 平均値を計算する。整数同士の割り算は小数点以下が切り捨てられて
42    しまう。そのため、一度 sum の値を avg に代入してfloat型の値に
43    変えた上で、avg を n で割る。
44    キャストを使って avg = (float)sum / n; としてもよい */
45    avg = sum;
46    avg /= n;
47
48    /* 表示する */
49    printf("最大値: %d\n", max);
50    printf("最小値: %d\n", min);
51    printf("平均値: %.3f\n", avg);
52 }
53
54 return 0;
55 }

```

## 実行結果例

values.txt の中身

39 2 -9 38 28 -37 18 -45 3 -24

```

fujie@DESKTOP-KVRJVM:~/work/kadai_example/06$ ./kadai1 < values.txt
最大値: 39
最小値: -45
平均値: 1.300

```

念のためExcelで確認 →

	39
	2
	-9
	38
	28
	-37
	18
	-45
	3
	-24
最大	39
最小	-45
平均	1.3

※values.txtが空のときは何も表示せずに終了(正しい動作)

```

fujie@DESKTOP-KVRJVM:~/work/kadai_example/06$ ./kadai1 < values.txt
fujie@DESKTOP-KVRJVM:~/work/kadai_example/06$

```

## 別解

```

1 /* 第6回 課題1 */
2 #include <stdio.h>
3
4 int main(void) {
5     /* rはscanfの戻り値を受け取るための変数、
6     xはscanfで入力された値を受け取るための変数、 */
7     int r, x;
8     /* それぞれ、最大値、最小値、合計値、入力された回数を表す変数、
9     最大値は、int型の整数の中でもっとも小さな値にしておき、
10    最小値は、その逆にもっとも大きな値にしておく。
11    こうすることで、最初に読み込んだ値がかならず最大値、最小値となる。
12    合計値や読み込んだ回数は0で初期化しておく */
13    int max = -2147483648, min = 2147483647, sum = 0, n = 0;
14    /* 平均値は少数を含む可能性があるためfloat型にする */
15    float avg;
16
17    while(1) {
18        /* 値を読み込む。読み込めなかった場合はループを抜ける */
19        r = scanf("%d", &x);
20        if(r < 0)
21            break;
22        /* 読み込んだ値が最大値よりも大きければ、最大値を更新する */
23        if(x > max)
24            max = x;
25        /* 読み込んだ値が最小値よりも小さければ、最小値を更新する */
26        if(x < min)
27            min = x;
28        /* 合計値と読み込み回数を更新する */
29        sum += x;
30        n++;
31    }

```

```

32    /* 少なくとも1回以上読み込めていないと、不適切な値が表示されて
33    しまうため、ここでそれを判定する */
34    if(n > 0) {
35        /* 平均値を計算する。整数同士の割り算は小数点以下が切り捨てられて
36        しまう。そのため、一度 sum の値を avg に代入してfloat型の値に
37        変えた上で、avg を n で割る。
38        キャストを使って avg = (float)sum / n; としてもよい */
39        avg = sum;
40        avg /= n;
41
42        /* 表示する */
43        printf("最大値: %d\n", max);
44        printf("最小値: %d\n", min);
45        printf("平均値: %.3f\n", avg);
46    }
47
48    return 0;
49 }
50 }

```

# 第6回

## 課題

- 課題2 for文またはwhile文による繰り返しで模様を描くプログラムをdrawlibを用いて作成せよ。

- 線分の端点の座標や、円の半径などを規則的に変化させて描画するとそれだけで模様になる
- 変化のさせ方によって模様も変わってくる
  - ・ 等差数列的にするか、等比数列的にするか

pattern.c

提出ファイルは「学籍番号\_kadai2.c」という名前にすること(例: 20C1981\_kadai2.c)

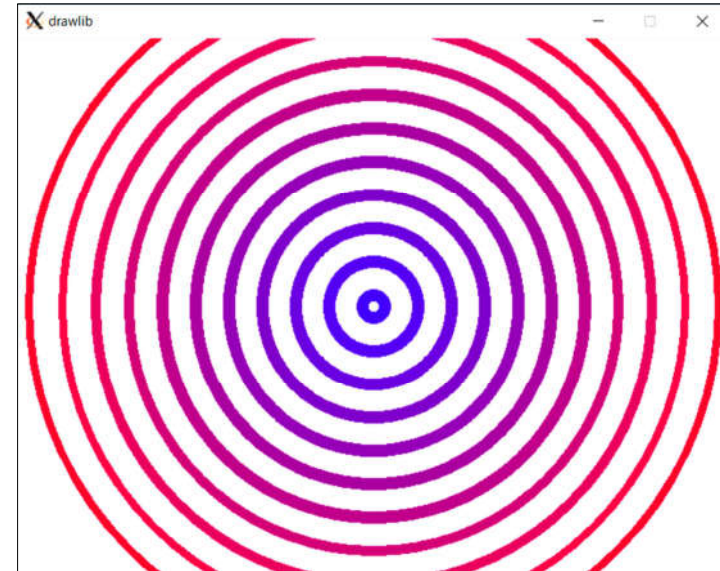


```

1 /* 第6回 課題2 */
2 #include <drawlib.h>
3
4 int main(void) {
5     int i;
6
7     dl_initialize(1.0);
8
9     /* ループさえあれば何でもOKなので、これはあくまで1例です */
10    for (i = 0; i <= 10; i++) {
11        /* iの値 (0~10) によって変化する円を描画します。
12         320, 240が円の中心座標ですが、これは画面の中心になります。
13         10 + i * 30が半径です。10~310を30刻みで大きくなっていきます。
14         DL_RGBは色を指定するマクロ (関数のようなもの) です。
15         赤の成分は55~255を20刻みで大きくなります。
16         青の成分は255~55を20刻みで小さくなります。
17         したがって、青から赤に徐々に変化していきます。
18         ちなみに、色の成分は255が最大値です。
19         線の太さは15から5まで1ずつ変化します。だんだん細くなります。
20         最後の0は、塗りつぶさないという意味です */
21        dl_circle(320, 240,
22                  10 + i * 30,
23                  DL_RGB(55 + 20 * i, 0, 255 - 20 * i),
24                  15 - i, 0);
25    }
26
27    while (1) {
28        dl_wait(1.0);
29    }
30
31    return 0;
32 }

```

## 実行結果例



## 第7回

### ■ 別途配布

## 第8回

```

1 #include <drawlib.h>
2 #include <stdio.h>
3
4 int main(void) {
5     /* --- 変数宣言 --- */
6     float wait_time = 0.01; /* drawlibの待機時間 */
7
8     /* ボール関係の変数 */
9     float bx, by; /* ボールのX座標, Y座標 */
10    float bvx, bvy; /* ボールのX方向の速度, Y方向の速度 */
11    float bay, bvy0; /* ボールのY方向の加速度, 初速 */
12    int br = 15; /* ボールの半径 */
13
14    /* ラケット (バー) 関係の変数 */
15    int bary = 440; /* バーの位置 (Y座標) */
16    int barh = 20; /* バーの高さ */
17    int bardiv = 7; /* バーの幅を画面の何分割にするか */
18    int barw = DL_WIDTH / bardiv; /* バーの幅 */
19    int barx = barw * (bardiv / 2); /* バーの位置 (X座標) */
20
21    /* スコア関係の変数 */
22    int score = 0; /* スコア */
23    int sx = DL_WIDTH / 2 - 120, sy = 50; /* スコアの表示位置 (X座標, Y座標) */
24    char sscore[10]; /* スコアを文字列化するための文字列 */
25
26    /* 制御関係の変数 */
27    int start = 0; /* 開始, 終了画面のループ判定用 */
28    int t, k, x, y; /* dl_get_event用 */
29
30    /* --- 未決定のボール関係の変数を設定 --- */
31    bx = 0.0; /* ボールの初期位置 (X座標) */
32    by = bary; /* ボールの初期位置 (Y座標) */
33
34    bay = 0.15; /* ボールのY方向の加速度 */
35    bvy0 = -10.0; /* ボールのY方向の初速 */
36    bvy = bvy0; /* ボールの速度 (Y方向) */
37    bvx = 1.0; /* ボールの速度 (X方向) */
38
39    dl_initialize(1.0);
40
41    /* --- 開始画面の表示 --- */
42    dl_stop();
43    dl_clear(DL_C("black"));
44    dl_text("PUSH 'F' to start", 160, 220, 1.0, DL_C("white"), 1);
45    dl_resume();
46
47    /* --- Fキーが押されるまで待機 --- */
48    start = 0;
49    while(start == 0) {
50        while(dl_get_event(&t, &k, &x, &y)) {
51            if (t == DL_EVENT_KEY) {
52                if (k == 'F') {
53                    start = 1;
54                }
55            }
56        }
57        dl_wait(wait_time);
58    }
59
60    /* --- メインループ --- */
61    while(1) {
62        /* 入力キーの処理 */
63        while(dl_get_event(&t, &k, &x, &y)) {
64            if (t == DL_EVENT_KEY) {
65                /* Jキーが押されたらバーのX座標を右にずらす */
66                if (k == 'J') {
67                    barx += barw;
68                }
69                /* Fキーが押されたらバーのX座標を右にずらす */
70                if (k == 'F') {
71                    barx -= barw;
72                }
73            }
74        }
75    }

```

## 第8回(つづき)

```

76  /* バーのX座標の境界処理(画面外に出ないように) */
77  if (barx < 0)
78      barx = 0;
79  if (barx + barw >= DL_WIDTH)
80      barx = DL_WIDTH - barw;
81
82  /* ボールの移動処理 */
83  bvy += bay; /* Y方向の速度に加速度を加える */
84  by += bvy; /* Y座標にY方向の速度を加える */
85  bx += bvx; /* X座標にX方向の速度を加える */
86
87  /* ボールの境界処理(左右) */
88  if (bx - br < 0) {
89      bvx *= -1.0; /* 速度の正負を反転させる */
90      bx = br; /* 位置の微調整 */
91  }
92  if (bx + br > DL_WIDTH) {
93      bvx *= -1.0; /* 速度の正負を反転させる */
94      bx = DL_WIDTH - br; /* 位置の微調整 */
95  }
96
97  /* バーとの当たり判定 */
98  if (by + br > bary && bx > barx && bx < barx + barw) {
99      bvy = bvy * 0; /* Y方向の速度は初速に戻す */
100     by = bary - br; /* 位置の微調整 */
101     score += 1; /* スコアを増やす */
102 }
103
104 /* ゲームオーバー判定 */
105 /* Y方向の画面外に出たらゲームオーバー */
106 if (by + br >= DL_HEIGHT) {
107     break;
108 }
109
110 /* 描画処理 */
111 dl_stop();
112 dl_clear(DL_C("black"));
113 dl_circle((int)bx, (int)by, br, DL_C("blue"), 1, 1);
114 dl_rectangle(barx, bary, barx + barw, bary + barh, DL_C("red"), 1, 1);
115 sprintf(sscore, "%5d", score);
116 dl_text(sscore, sx, sy, 2.0, DL_C("white"), 2);
117 dl_resume();
118 dl_wait(wait_time);
119 }
120
121 /* --- 終了画面の表示 --- */
122 dl_stop();
123 dl_clear(DL_C("black"));
124 sprintf(sscore, "%5d", score);
125 dl_text(sscore, sx, sy, 2.0, DL_C("white"), 2);
126 dl_text("GAME OVER", 160, 220, 1.0, DL_C("white"), 1);
127 dl_text("push 'F' to quit", 160, 300, 1.0, DL_C("white"), 1);
128 dl_resume();
129
130 /* --- Fキーが押されるまで待機 --- */
131 start = 0;
132 while(start == 0) {
133     while (dl_get_event(&t, &k, &x, &y)) {
134         if (t == DL_EVENT_KEY) {
135             if (k == 'f') {
136                 start = 1;
137             }
138         }
139     }
140     dl_wait(wait_time);
141 }
142
143 return 0;
144 }

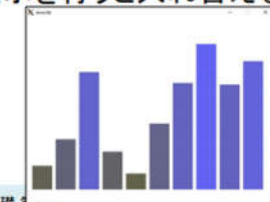
```

## 第9回

### 課題

#### ■ 課題1

- 配布した `kadai1.c` を改変し、背の低い順にバーが整列するようなプログラムにせよ。
  - コード内の変更可否の指示にしたがって変更する部分に注意すること
- 入れ替えの様子が分かるように注意すること  
(ソートが済んでから表示を行うと入れ替える様子がわからない)



プログラミング基礎

## 解答例(バブルソート)

```

15  /* --- ここから変更可 --- */
16  for (i = 0; i < 9; i++) {
17      for (j = 0; j < 9 - i; j++) {
18          if (values[j] > values[j + 1]) {
19              /* j番目とj+1番目を入れ替える */
20              s = values[j];
21              values[j] = values[j + 1];
22              values[j + 1] = s;
23          }
24      }
25  }
26  /* --- ここまで変更可 --- */
27
28  ...
29
30  /* --- ここから変更可 --- */
31  }
32  }
33  /* --- ここまで変更可 --- */

```

## 解答例(選択ソート)

```

15  /* --- ここから変更可 --- */
16  for (i = 0; i < 9; i++) {
17      for (j = 0; j < 9 - i; j++) {
18          if (values[j] > values[9 - i]) {
19              /* j番目と9-i番目を入れ替える */
20              s = values[j];
21              values[j] = values[9 - i];
22              values[9 - i] = s;
23          }
24      }
25  }
26  /* --- ここまで変更可 --- */
27
28  ...
29
30  /* --- ここから変更可 --- */
31  }
32  }
33  /* --- ここまで変更可 --- */

```

## 第9回

### 課題

#### ■ 課題2

- 10個の整数を標準入力から読み込み、昇順(小さい順)に並べ替えて表示するプログラムを書け。  
例えば,  
23, 94, 8, 19, 83, 34, 53, 85, 50, 59  
という数値が与えられた場合は  
8, 19, 23, 34, 50, 53, 59, 83, 85, 94  
と表示する

```
1 /* 第9回 課題2 */
2 #include <stdio.h>
3
4 int main(void) {
5     int i, j, s, x[10];
6
7     /* 10個の整数を読み込み、配列xの各要素に代入する */
8     for (i = 0; i < 10; i++) {
9         scanf("%d", &x[i]);
10        printf("%d ", x[i]);
11    }
12    printf("\n");
13
14    /* xをソートする(バブルソート) */
15    for (i = 0; i < 9; i++) {
16        for (j = 0; j < 9; j++) {
17            if (x[j] > x[j + 1]) {
18                s = x[j];
19                x[j] = x[j + 1];
20                x[j + 1] = s;
21            }
22        }
23    }
24 }
```

```
25 /* xを表示する */
26 for (i = 0; i < 10; i++) {
27     printf("%d ", x[i]);
28 }
29 printf("\n");
30
31 return 0;
32 }
```

#### 実行例

values.txt の中身 23 94 8 19 83 34 53 85 50 59

```
fujie@yellow:~/work/answers/09$ ./kadai2 < values.txt
23 94 8 19 83 34 53 85 50 59
8 19 23 34 50 53 59 83 85 94
```

## 第10回

### 課題

- 以下の要件を満たすプログラムを作成せよ。
    - 標準入力から20文字以内の文字列(半角英数字)をchar型の配列に読み込む
    - 読み込んだ配列の中身(文字列)を標準出力に出力する
    - 読み込んだ配列の要素を、前から順番にポインタptrで走査し、小文字を大文字、大文字を小文字に変更する
    - 変更した結果を標準出力に表示する。
- ※標準入力からの入力はscanf関数を使ってよい  
※標準出力への出力はprintf関数を使ってよい  
※大文字小文字の判定や変換に標準関数は使わないこと

締切は6月29日(火) 23:59



```

1 #include <stdio.h>
2
3 int main(void) {
4     char str[21];
5     char *ptr;
6
7     printf("20文字以内の文字列を入力してください: ");
8     scanf("%20s", str);
9
10    printf("%s\n", str);
11
12    ptr = str;
13    while (*ptr != '\0') {
14        if (*ptr >= 'a' && *ptr <= 'z') {
15            *ptr += 'A' - 'a';
16        } else if (*ptr >= 'A' && *ptr <= 'Z') {
17            *ptr += 'a' - 'A';
18        }
19        ptr++;
20    }
21
22    printf("%s\n", str);
23
24    return 0;
25 }

```

# 第11回 課題1

## 課題1

- 問題: 次の要件を満たすプログラムを作成し、ソースコード(kada11.c)を提出してください。
  - double型の引数x1, y1, x2, y2をとり、2点(x1, y1)と(x2, y2)間の距離を求めてdouble型で戻す関数distが定義されている
    - dist関数の定義内では標準関数sqrtを使ってよい
  - main関数内では次の処理をする。
    - (1) 3点の座標を標準入力から得る
    - (2) その3点を作る三角形が、鋭角三角形、鈍角三角形、直角三角形のどれか、あるいは三角形を形作らないかを判定する
    - (3) 判定結果を標準出力に出力する

```

1 #include <stdio.h>
2 #include <math.h>
3
4 double dist(double x1, double y1, double x2, double y2);
5
6 int main(void) {
7     double x1, y1, x2, y2, x3, y3;
8     double d1, d2, d3, s;
9     double sd1, sd2, sd3;
10
11     printf("X1: ");
12     scanf("%lf", &x1);
13     printf("Y1: ");
14     scanf("%lf", &y1);
15     printf("X2: ");
16     scanf("%lf", &x2);
17     printf("Y2: ");
18     scanf("%lf", &y2);
19     printf("X3: ");
20     scanf("%lf", &x3);
21     printf("Y3: ");
22     scanf("%lf", &y3);
23
24     d1 = dist(x1, y1, x2, y2);
25     d2 = dist(x2, y2, x3, y3);
26     d3 = dist(x3, y3, x1, y1);
27
28     if (d2 > d1) {
29         s = d1;
30         d1 = d2;
31         d2 = s;
32     }
33     if (d3 > d1) {
34         s = d1;
35         d1 = d3;
36         d3 = s;
37     }
38
39     sd1 = d1 * d1;
40     sd2 = d2 * d2;
41     sd3 = d3 * d3;
42
43     if (d1 > d2 + d3 - 0.000001 && d1 < d2 + d3 + 0.000001) {
44         printf("三角形にならない\n");
45     } else if (sd1 < sd2 + sd3 - 0.000001) {
46         printf("鋭角三角形\n");
47     } else if (sd1 > sd2 + sd3 + 0.000001) {
48         printf("鈍角三角形\n");
49     } else {
50         printf("直角三角形\n");
51     }
52     return 0;
53 }
54
55 double dist(double x1, double y1, double x2, double y2) {
56     double dx, dy, d;
57     dx = x2 - x1;
58     dy = y2 - y1;
59     d = sqrt(dx * dx + dy * dy);
60     return d;
61 }

```

# 第11回 課題2

## 課題2

- 問題: 次の要件を満たすプログラムを作成し、ソースコード(kada12.c)を提出してください。
  - 2つの文字列の辞書順を調べる関数
    - int strcmp(char \*str1, char \*str2);
    - を定義する。strcmpの戻り値は、辞書順でstr1がstr2より先の場合は-1, str1とstr2が等しければ0, str1がstr2より後の場合は1とする。
  - 文字列配列を受け取って辞書順(昇順)にソートする関数
    - void dictsort(char \*\*, int);
    - を定義する。第1引数が文字列配列の先頭ポインタで、第2引数が配列の要素数である。定義の中で上記で作成したstrcmp関数を使うこと。

## 課題(つづき)

- main関数では、
 

```
char *flowers[] = {"cherry", "ume", "tulip", "dandelion", "peach", "marigold", "azalea", "orchid", "lily", "carnation", "pansy", "hyacinth", "iris"};
```

と初期化された文字列配列 flowers の要素を、上記で作成した dictsort を使って辞書順にソートし、その結果を標準出力に表示する。
- 注意
  - 文字列操作の標準関数を使わないこと (strcmpやソートをする関数は標準関数として使えるものがあるが、今回は使わないこと)

```

1 #include <stdio.h>
2
3 int strcmp(char *, char *);
4 void dictsort(char **, int);
5
6 int main(void) {
7     char *flowers[] = {"cherry", "ume", "tulip", "dandelion",
8         "peach", "marigold", "azalea", "orchid", "lily",
9         "carnation", "pansy", "hyacinth", "iris"};
10    int n = 13, i;
11
12    dictsort(flowers, n);
13
14    for (i = 0; i < n; i++) {
15        printf("%s\n", flowers[i]);
16    }
17
18    return 0;
19 }
20

```

```

21 int strcmp(char *str1, char *str2) {
22     char *ptr1, *ptr2;
23
24     ptr1 = str1;
25     ptr2 = str2;
26     while (*ptr1 == *ptr2) {
27         if (*ptr1 == '\0')
28             return 0;
29         ptr1++;
30         ptr2++;
31     }
32     if (*ptr1 < *ptr2)
33         return -1;
34     else
35         return 1;
36 }
37
38 void dictsort(char **sl, int n) {
39     int i, j;
40     char *s;
41
42     for (i = 0; i < n - 1; i++) {
43         for (j = 0; j < n - 1 - i; j++) {
44             if (strcmp(sl[j], sl[j + 1]) > 0) {
45                 s = sl[j];
46                 sl[j] = sl[j + 1];
47                 sl[j + 1] = s;
48             }
49         }
50     }
51     return;
52 }

```