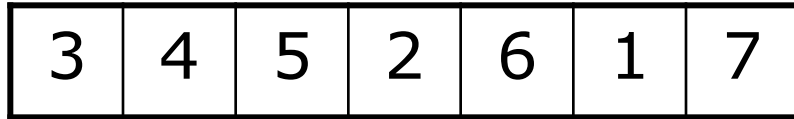


プロセツシング 入門3

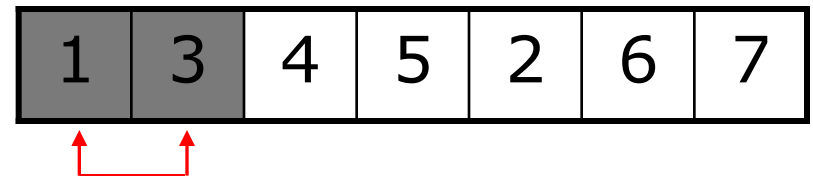
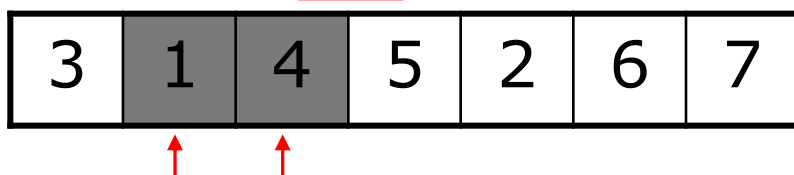
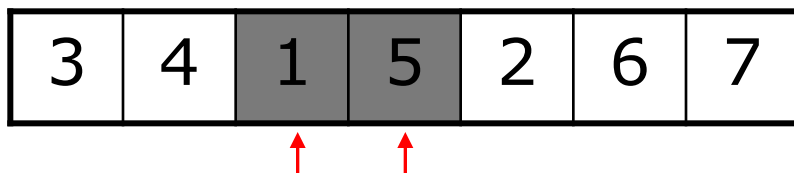
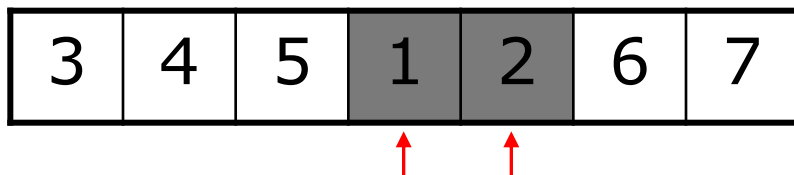
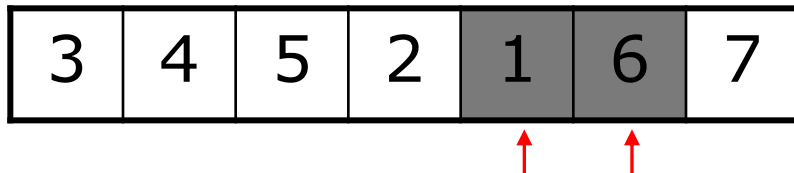
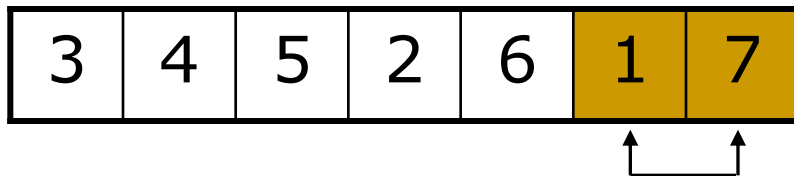
初歩のプログラミング

単純交換ソート(バブルソート)

次に示す数字の並びを昇順に並べ替えることを考える。

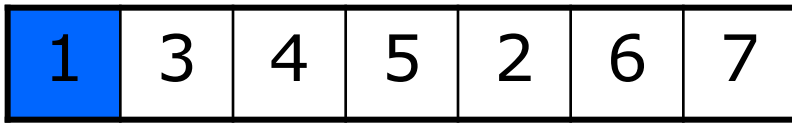


(1ラウンド目)

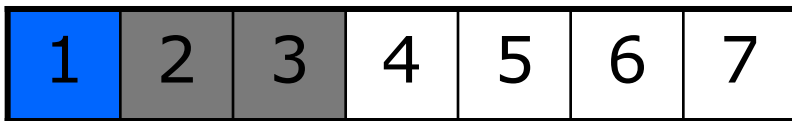
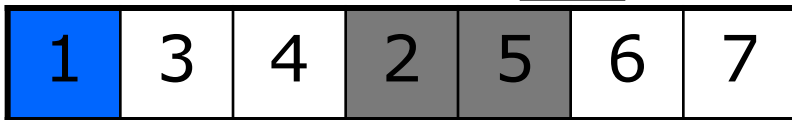
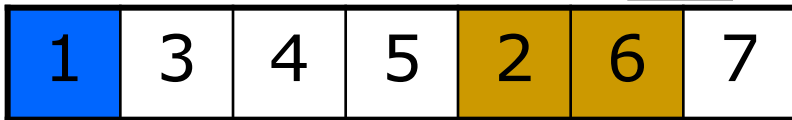
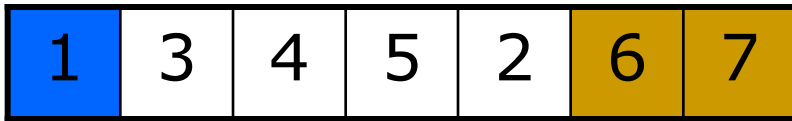


↑ ↑ 比較するが交換しない

↑ ↑ 比較して交換する



(2ラウンド目)

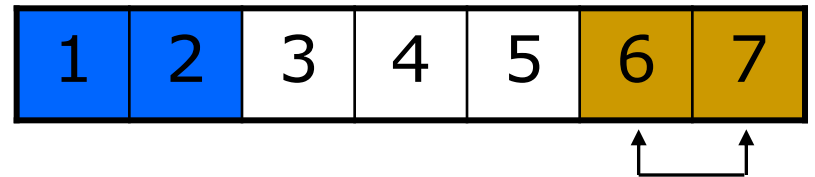


液体中の気泡(液体より軽い)が上に上がってゆくイメージから**バブルソート**という名がついています。

バブルソートは隣り合う2つのデータで左の方が大きければ、入れ替え、そうでなければ入れ替えを行いません。

1ラウンドごとに左から小さい順に要素が確定しますから、ラウンドが進むにつれ、比較の回数は少なくなります。

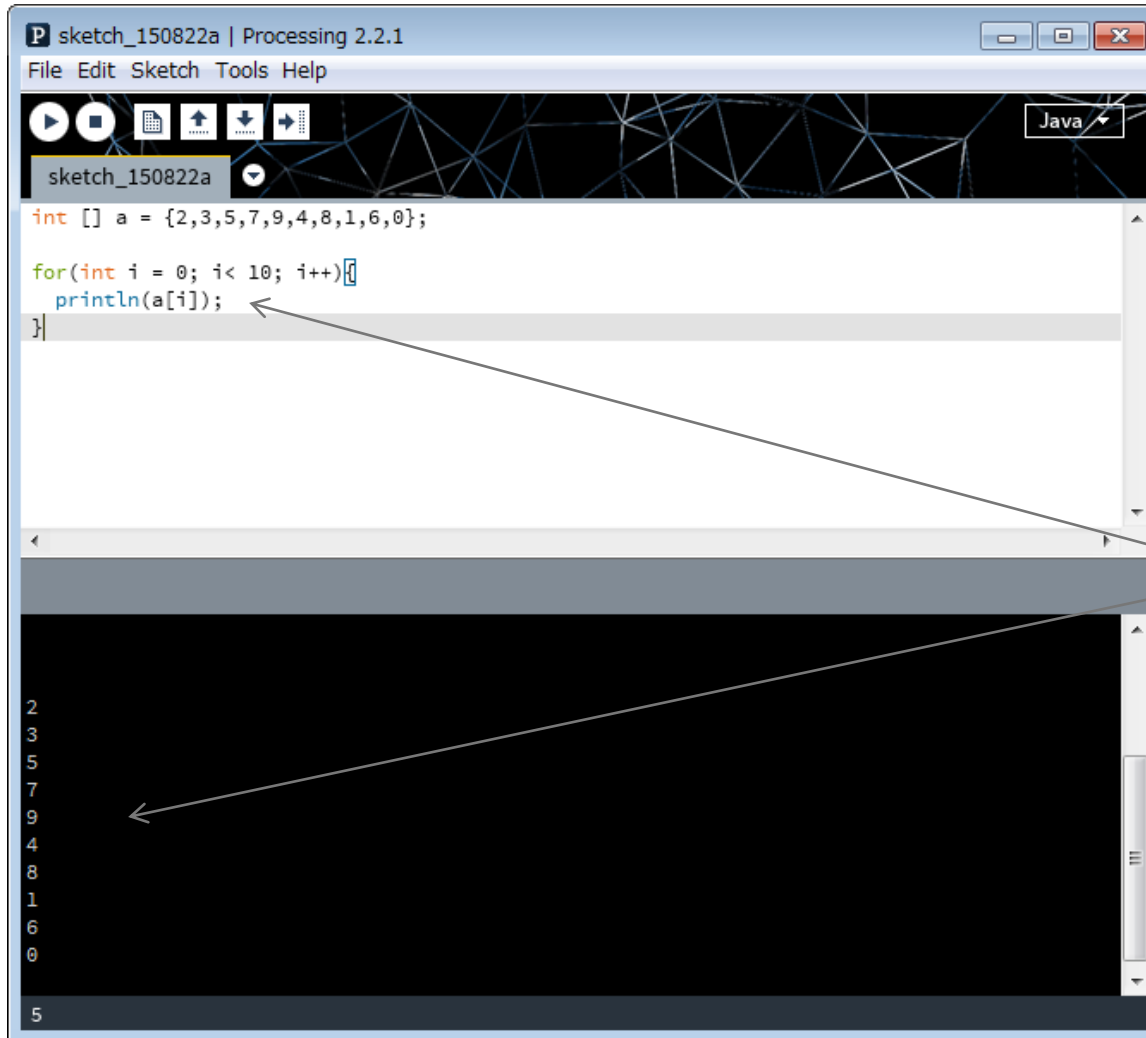
(3ラウンド目)



つづく...

配列

データは配列に格納します。



The screenshot shows a Processing IDE window titled "sketch_150822a | Processing 2.2.1". The code in the editor is as follows:

```
int [] a = {2,3,5,7,9,4,8,1,6,0};  
  
for(int i = 0; i < 10; i++){  
  println(a[i]);  
}
```

The output window at the bottom shows the following values printed on separate lines:

```
2  
3  
5  
7  
9  
4  
8  
1  
6  
0  
  
5
```

Arrows from the callouts point to the `println(a[i]);` line in the code and the output window.

この例では
a[0]=1, a[1]=3, a[2]=5, a[9]=0
のように格納されます。
データの個数は10個ですが、a[10]=0
ではないことに注意してください。

println(x)はxの値を
下の黒い画面に出力する
関数です

テキスト版バブルソート

```
P b_sort_text | Processing 2.2.1
File Edit Sketch Tools Help

b_sort_text

int [] a = {2,3,5,7,9,4,8,1,6,0};

for(int i = 0; i < 10; i++){
  println(a[i]);
}

print("++++++\n");

for(int i = 0; i < a.length-1; i++){
  for (int j = a.length-2; j >= i; j--) {
    if(a[j+1] < a[j]){
      int tmp = a[j];
      a[j] = a[j+1];
      a[j+1] = tmp;
    }
  }
}

for(int i = 0; i < 10; i++){
  println(a[i]);
}
```

Done Saving.

```
2
3
5
7
9
4
8
1
6
0
++++++
0
1
2
3
4
5
6
7
8
9
7
```

← ラウンド目を表示します

a.lengthは配列aの大きさを表します。
すなわち値は10です。
数字の10を書いても構いません。

j番目とj+1番目を比較する訳ですが
最大のjは8です。 a.length-2は8なので
8と書いてしまっても構いません。

入れ替え後の表示をしています。

問1
配列aに入っている値を20個に変更して実行してみよ。

問2
データが大きい順に出力されるよう変更してみよ

関数(雰囲気があればよい)

```
sketch_150822a | Processing 2.2.1
File Edit Sketch Tools Help
sketch_150822a
void setup(){
  size(600,200);
  noLoop(); ←
}

void draw(){
  for(int i = 0; i < 5; i++){
    drawcircle(i); ←
  }
}

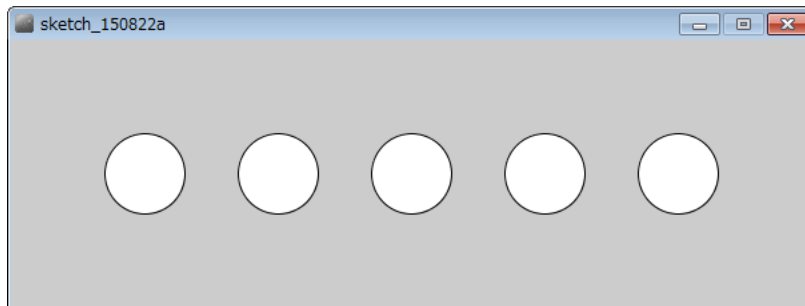
void drawcircle(int i){
  ellipse(100 + 100 * i, 100, 60, 60);
}

15
```

今回はdraw()は1回実行すればよいので、noLoop()とします。

値iを関数drawcircle(int i)に渡します。呼び出された関数は、iの値に応じた直径60の円を描きます。

問3
関数drawcircle(int i, int r)に変更してrで円の直径を変えることができるようにせよ。



関数を使ったテキスト版バブルソート

```
sketch_150822a | Processing 2.2.1
File Edit Sketch Tools Help

sketch_150822a

int [] a = {2,3,5,7,9,4,8,1,6,0};

void setup(){
  noLoop();
}

void draw(){
  for(int i = 0; i < 10; i++){
    println(a[i]);
  }
  print("+++++\n");
  for(int i = 0; i < a.length-1; i++){
    bubble(i);
  }
  for(int i = 0; i < 10; i++){
    println(a[i]);
  }
}

void bubble(int i){
  for (int j = a.length-2; j >= i; j--) {
    if(a[j+1] < a[j]){
      int tmp = a[j];
      a[j] = a[j+1];
      a[j+1] = tmp;
    }
  }
}

2
3
5
-
3
```

問4

void bubble(int i, int j)に関数を変更し、j=0ならば、データが小さい順に出力されj=1ならば、データが大きい順に出力されるようにしてみよ

2つの荷電粒子のつくる電場のシミュレーション

sketch_160815b

```
float Q1_f=50*20000,Q2_f=-25*20000;<----- 2つの荷電粒子の電荷
```

```
void setup(){  
  size(800,800);<----- ウィンドウサイズ=800×800  
  noLoop();  
}
```

```
void draw(){  
  float x,y;  
  float r1,r2;  
  //arrow(10,200,30,40);  
  fill(255,255,0);<----- 黄色で荷電粒子を描く  
  ellipse(280, 400, 15, 15);<----- 荷電粒子の描画(280,400),  
  ellipse(520, 400, 15, 15);<----- (520,400)に配置  
  line(0,400,800,400);<----- x軸y軸の描画  
  line(400,0,400,800);
```

```
  for(int i = 0; i < 20; i++){  
    for(int j = 0; j < 20; j++){  
      if((40*i == 280 && 40*j == 400) || (40*i == 520 && 40*j == 400)) continue;  
      r1 = sqrt((float)(pow((40*i-280),2)+pow((40*j-400),2)));  
      r2 = sqrt((float)(pow((40*i-520),2)+pow((40*j-400),2)));  
      x = Q1_f*((float)(40*i-280)/(r1*r1*r1))  
        + Q2_f*((float)(40*i-520)/(r2*r2*r2));  
      y = Q1_f*((float)(40*j-400)/(r1*r1*r1))  
        + Q2_f*((float)(40*j-400)/(r2*r2*r2));  
      //println(x,y);  
      if(x*x+y*y>35*35){<----- 矢印が大きくなりすぎる場合の処置  
        x = 35*x/sqrt(x*x+y*y);  
        y = 35*y/sqrt(x*x+y*y);  
      }  
      arrow(40*i,40*j,(int)(40*i+x),(int)(40*j+y));  
    }  
  }  
}
```

座標 $(40i, 40j)$ における電場の x, y 成分(E_x, E_y)を求めている。

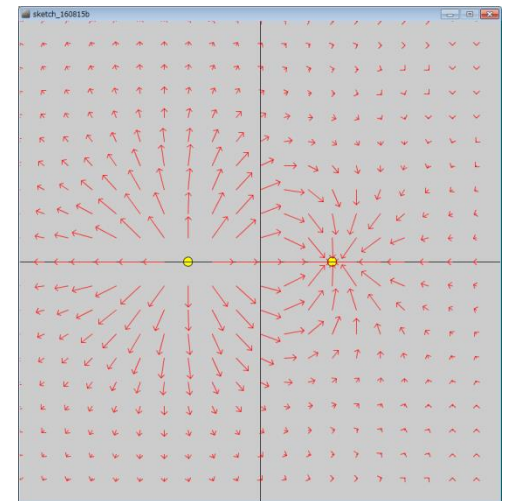
$$r_1 = \sqrt{(40i - 280)^2 + (40j - 400)^2}$$

$$r_2 = \sqrt{(40i - 520)^2 + (40j - 400)^2}$$

とすると

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = k \frac{Q_1}{r_1^2} \frac{1}{r_1} \begin{pmatrix} 40i - 280 \\ 40j - 400 \end{pmatrix} + k \frac{Q_2}{r_2^2} \frac{1}{r_2} \begin{pmatrix} 40i - 520 \\ 40j - 400 \end{pmatrix}$$

2つの荷電粒子上の電場は定義できない



2つの荷電粒子のつくる電場のシミュレーション

```
void arrow(int x1, int y1, int x2, int y2) {  
    stroke(255,0,0);  
    line(x1, y1, x2, y2);  
    pushMatrix();  
    translate(x2, y2);  
    float a = atan2(x1-x2, y2-y1);  
    rotate(a);  
    line(0, 0, -5, -5);  
    line(0, 0, 5, -5);  
    popMatrix();  
}
```

← 始点 (x_1, y_1) , 終点 (x_2, y_2) のベクトルを描く
関数と理解しておけばよい

問4 電荷の大きさ $Q1_f$ と $Q2_f$ を変えてシミュレーションしてみよ

問5 荷電粒子の配置を $(320,400)$ と $(480,400)$ に変更してみよ

このシミュレーションは「基礎電磁気学演習」
でやった(5)番にあたる。納得したことでしょう。



96. <電気力線>
xy平面上において、距離 l [m] だけ離れた 2 点 A, B に電荷を固定したときの電気力線について考える。点 A の座標を $(-\frac{l}{2}, 0)$, 点 B の座標を $(\frac{l}{2}, 0)$ として以下の設問に答えよ。

[A] 点 A, 点 B に等しい正電荷 Q [C] を置いた場合を考える。
(1) xy 平面上の電気力線のようすを、向きも含めて図示せよ。
(2) Q が 5.0×10^{-12} C, l が 6.0×10^{-2} m とする。y 軸上で原点から 4.0×10^{-3} m だけ離れた点に静かに置いた大きさの無視できる荷電粒子が、無限遠方に達したときの速度を求めよ。ただし、荷電粒子の電荷を 1.6×10^{-19} C, 質量を 9.0×10^{-31} kg とする。また、クーロンの法則の比例定数を 9.0×10^9 N·m²/C² とする。

[B] 点 A, 点 B にそれぞれ Q [C], $-\frac{Q}{2}$ [C] の電荷を置いた場合を考える。ただし、 Q は正とする。
(1) 電位が 0 (無限遠方と同じ) となる点 (x, y) が満たす方程式を求めよ。それは xy 平面上でどのような図形を表すか。
(2) x 軸上の点 P に電荷を置いたとき、それにはたらく力が 0 になった。そのような点 P の座標を求めよ。
(3) 点 B を中心とする円周上で、電位が最も低い点は x 軸上 (ただし $x > \frac{l}{2}$) にある。その理由を説明せよ。
(4) 点 A を出た電気力線は、一部は点 B に、一部は無限遠方に達する。線分 AB となす角度 θ で点 A を出た電気力線が点 B に入るとき、 θ がとりうる範囲を理由とともに答えよ。ただし、電気力線の影響を無視してよい。
(5) 設問 [B] (1) から設問 [B] (4) の結果を参考にして、xy 平面上の電気力線のようすを、向きも含めて特徴がわかるように図示せよ。なお、図には点 A, 点 B, 点 P の位置をそれぞれ示すとともに、設問 [B] (1) で求めた図形を点線でかき加えよ。