

総合型選抜プログラミング実技試験  
サンプル問題

日本大学文理学部情報科学科

1. 次の各問のプログラムを作成せよ。

- (1) 夏季オリンピックは、5つの例外（1906年アテネ大会開催、1916年ベルリン大会中止、1940年ヘルシンキ大会中止、1944年ロンドン大会中止、2020年東京大会2021年へ延期）を除き、1896年の第一回大会から4年毎に開催されている。さて、西暦  $Y$  年との差が最も小さい夏季オリンピック開催年（予定を含む）をコンソールに表示するプログラムを作成せよ。なお、例えば  $Y = 1898$  に対する1896と1900の様に、該当する年が複数ある場合は、両方を表示するものとする。 $Y$ の値はプログラム先頭で与えるものとする。
- (2) 7000より大きく7500以下の7の倍数を、大きい方から順番にすべてコンソールに表示するプログラムを作成せよ。
- (3) 生年月日  $S$  を8桁の整数で表したものを  $X_0$ 、 $F(X)$  を  $X$  の各桁の和とする。自然数  $i$  について、 $X_i = F(X_{i-1})$  としたとき、 $X_i$  はやがて1桁の整数となる。言い換えると  $X_{k+1} = X_k$  となる  $k$  が存在し、このときの  $X_k$  を  $S$  の運命数と呼ぶ。

2020年10月17日の運命数の計算例

$$X_0 = 20201017$$

$$X_1 = 2 + 0 + 2 + 0 + 1 + 0 + 1 + 7 = 13$$

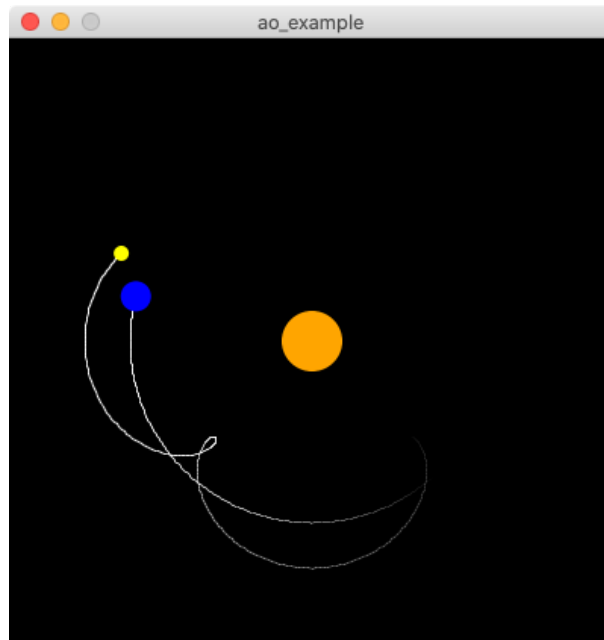
$$X_2 = 1 + 3 = 4$$

運命数を計算するプログラムを作成し、1999年9月29日 ( $X_0 = 19990929$ )の運命数をコンソールに表示せよ。ただし、1999年9月29日以外の生年月日についても正しく運命数を計算できるようにすること。

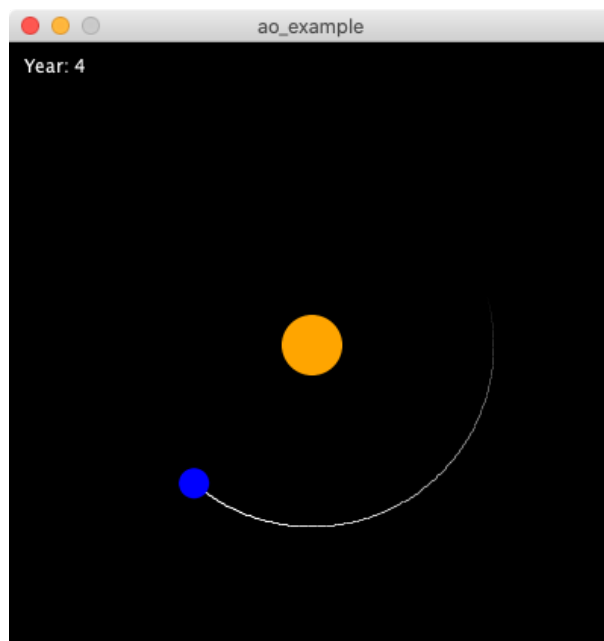
2. 事前公開プログラムは、地球が太陽の周りをまわる様子を模したアニメーションを描画する。太陽は画面の中心にオレンジ色の円で描かれ、地球は太陽を中心とした等速円運動をする青色の円として描かれる。また、地球の運動の軌跡が白線で描かれる。

次の指示に従って事前公開プログラムを拡張せよ。なお、提出するプログラムは、すべての拡張を行った最終的なもの1つだけでよい。

- (1) 地球の周りを等速円運動する月を追加せよ。地球と同様に月の軌跡も描くこと。



- (2) 天体の運動速度を、キーボードの上矢印キーを押したとき現在の倍に、下矢印キーを押したとき現在の半分に、スペースキーを押したとき最初速度に戻すようにせよ。
- (3) 地球が太陽の周りを周回した回数  $y$  を「Year:  $y$ 」の形式で画面に表示せよ。



- (4) 自分の自由な発想でプログラムを拡張せよ。どのような拡張を行ったかをソースコードの冒頭にコメントとして記述すること。

Java Mode

```
1 int n = 256;  
2 float sx, sy;
```

```

3 float sa = 40;
4 float [] ex = new float [n];
5 float [] ey = new float [n];
6 float ea = 20;
7 float er = 120;
8 float et = 0;
9 float edt = 0.01;
10 int repeat = 0;
11
12 void setup() {
13     size(400, 400);
14     sx = width / 2;
15     sy = height / 2;
16 }
17
18 void draw() {
19     background(0);
20
21     updatePosition();
22     drawSun();
23     drawEarth();
24
25     repeat += 1;
26 }
27
28 void drawSun() {
29     fill(255, 165, 0);
30     noStroke();
31     ellipse(sx, sy, sa, sa);
32 }
33
34 void drawEarth() {
35     for (int i = 1; i < min(n, repeat); ++i) {
36         stroke(255, 256 - i);
37         line(ex[i - 1], ey[i - 1], ex[i], ey[i]);
38     }
39     fill(0, 0, 255);
40     noStroke();
41     ellipse(ex[0], ey[0], ea, ea);
42 }
43
44 void updatePosition() {
45     for (int i = n - 1; i > 0; --i) {
46         ex[i] = ex[i - 1];
47         ey[i] = ey[i - 1];
48     }
49     ex[0] = sx + er * cos(et);
50     ey[0] = sy + er * sin(et);
51     et += edt;
52 }

```

### Python Mode

```

1 n = 256
2 sx = 0
3 sy = 0
4 sa = 40
5 ex = [0.0] * n
6 ey = [0.0] * n
7 ea = 20
8 er = 120
9 et = 0
10 edt = 0.01
11 repeat = 0
12
13 def setup():

```

```

14     global sx, sy
15     size(400, 400)
16     sx = width / 2
17     sy = height / 2
18
19     def draw():
20         global repeat
21         background(0)
22
23         updatePosition()
24         drawSun()
25         drawEarth()
26
27         repeat += 1
28
29     def drawSun():
30         fill(255, 165, 0)
31         noStroke()
32         ellipse(sx, sy, sa, sa)
33
34     def drawEarth():
35         for i in range(1, min(n, repeat)):
36             stroke(255, 256 - i)
37             line(ex[i - 1], ey[i - 1], ex[i], ey[i])
38             fill(0, 0, 255)
39             noStroke()
40             ellipse(ex[0], ey[0], ea, ea)
41
42     def updatePosition():
43         global et
44         for i in range(n - 1, 0, -1):
45             ex[i] = ex[i - 1]
46             ey[i] = ey[i - 1]
47         ex[0] = sx + er * cos(et)
48         ey[0] = sy + er * sin(et)
49         et += edt

```

#### p5.js Mode

```

1 let n = 256;
2 let sx, sy;
3 let sa = 40;
4 let ex = new Array(n);
5 let ey = new Array(n);
6 let ea = 20;
7 let er = 120;
8 let et = 0;
9 let edt = 0.01;
10 let iter = 0;
11
12 function setup() {
13     createCanvas(400, 400);
14     sx = width / 2;
15     sy = height / 2;
16 }
17
18 function draw() {
19     background(0);
20
21     updatePosition();
22     drawSun();
23     drawEarth();
24
25     iter += 1;
26 }
27

```

```
28 function drawSun() {
29   fill(255, 165, 0);
30   noStroke();
31   ellipse(sx, sy, sa, sa);
32 }
33
34 function drawEarth() {
35   for (let i = 1; i < min(n, iter); ++i) {
36     stroke(255, 256 - i);
37     line(ex[i - 1], ey[i - 1], ex[i], ey[i]);
38   }
39   fill(0, 0, 255);
40   noStroke();
41   ellipse(ex[0], ey[0], ea, ea);
42 }
43
44 function updatePosition() {
45   for (let i = n - 1; i > 0; --i) {
46     ex[i] = ex[i - 1];
47     ey[i] = ey[i - 1];
48   }
49   ex[0] = sx + er * cos(et);
50   ey[0] = sy + er * sin(et);
51   et += edt;
52 }
```