# アルゴリズム入門　共通問題（2018年度Aセメスター試験）

[ 試験日時：2018 年 1 月 24 日第 3 限（60 分），答案用紙：1 部，計算用紙：1 枚，持ち込み一切不可]

内容に関する質問は受け付けない．問題の記述が曖昧な場合は，適切な仮定をおいて回答し，どのような仮定をおいたのか明記せよ．

## 問題 1

与えられた 2 以上の整数 $n$ が素数かどうかを判定する関数を定義したい．以下の問に答えよ．

(1) 2 から $n-1$ までの全ての数で順に割り，割り切れなければ素数だと判定するアルゴリズムを isPrime1 関数として定義した．空欄を埋めよ．

```
def isPrime1(n):
  for i in range(   A   , n):
    if   B   == 0:
      return False
  return True
```

(2) isPrime1 関数に現れる 2 つの return のインデントが異なる理由を 2 ～ 3 行程度で説明せよ．

(3) $n-1$ まで試す必要はなく，$\frac{n}{2}$ まで調べれば十分であることに気づいた．これに基づき isPrime1 関数の range( A , n) を range( A , n / 2 + 1) と変更して isPrime1(15) を実行したところ，以下のエラーメッセージが出力された．このエラーの理由を 1 ～ 2 行程度で説明せよ．

```
TypeError: 'float' object cannot be interpreted as an integer
```

(4) $n$ より小さな素数全てで順に割り，割り切れなければ素数だと判定するアルゴリズムを考えた．これを実現するため，まずはそれより小さな素数が配列 primes として与えられたとき，$n$ が素数かどうかを判定する関数 isPrime_sub を定義した．空欄を埋めよ．

```
def isPrime_sub(n, primes):
  for i in range(0, len(primes)):
    if   C   == 0:
      return False
  return True
```

(5) 関数 isPrime_sub を用いて素数判定を行う関数 isPrime2 の空欄を埋めよ．なお，append は与えられた数を配列の末尾要素として追加する関数である．

```
def isPrime2(n):
  primes = []
  for i in range(   D   , n):
    if   E   :
      primes.append(i)
  return isPrime_sub(n, primes)
```

(6) isPrime1 関数と isPrime2 関数を漸近計算量の観点から 2 ～ 3 行程度で比較せよ．

## 問題 2

自由落下する物体の時刻 $t$ における高さ $y(t)$ と速度 $v(t)$ を計算するプログラムを考える．重力加速度を $g > 0$ とすると，以下の微分方程式が成り立つ．

$$\frac{\mathrm{d}}{\mathrm{d}t}y(t) = v(t), \quad \frac{\mathrm{d}}{\mathrm{d}t}v(t) = -g \tag{a}$$

ここで，時刻 $t = 0$ における高さと初速をそれぞれ $y(0) = 0, v(0) = v_0$ とすると，以下の関係式が成り立つ．以下の問に答えよ．

$$y(t) = -\frac{1}{2}gt^2 + v_0 t, \quad v(t) = -gt + v_0 \tag{b}$$

(1) 重力加速度 g，初速 v0 のとき，時刻 t における高さ y と速度 v を (b) 式に従い求める関数 exact(g, v0, t) は次のようになる．空欄を埋めよ．

```
def exact(g, v0, t):
  y = [ A ]
  v = [ B ]
  return [y, v]
```

(2) 重力加速度 g，初速 v0 のとき，(a) 式を差分化し，時刻 t までを n ステップに分割し，高さ y と速度 v を計算する関数 numeric(g, v0, t, n) は，次のようになる．空欄を埋めよ．

```
def numeric(g, v0, t, n):
  dt = t / n
  y = 0
  v = [ C ]
  for i in range(0, n):
    y = [ D ]
    v = v - g * dt
  return [y, v]
```

(3) 関数 numeric(g, v0, t, n) におけるステップ数 n の効果を測るために，compare_y(v0) 関数を定義した．この関数は，初速 v0 を引数にとり，重力加速度 g が 10.0，時刻 t が 1.0，n が 10，$10^2$，$10^3$，$10^4$，$10^5$，$10^6$ の時の numeric(g, v0, t, n) と exact(g, v0, t) が求める高さの差の絶対値を配列として返す．また、numeric と exact が求める速度の差の絶対値の配列を compare_y と同様にして求める関数 compare_v も定義した．
初速 v0 を 10 として得られた以下の実行結果から，関数 numeric(g, v0, t, n) におけるステップ数の効果として，どのようなことが言えるだろうか．2 行程度で述べよ．丸め誤差，桁落ち，情報落ち，打切り誤差などとの関係があれば述べること．

```
> compare_y(10)
[0.5, 0.050000000000013145, 0.005000000000102922,
 0.0005000000002777227, 5.0000008545936225e-05, 5.000154284395819e-06]
```

(4) 初速 v0 を $10^{11}$ としたところ，以下の実行結果が得られた．この結果から，関数 numeric(g, v0, t, n) について，どのようなことが言えるだろうか．2 行程度で述べよ．丸め誤差，桁落ち，情報落ち，打切り誤差などとの関係があれば述べること．

```
> compare_y(10 ** 11)
[0.5000152587890625, 0.049713134765625, 0.0077362060546875,
 0.035003662109375, 0.3404998779296875, 3.268280029296875]
> compare_v(10 ** 11)
[0.0, 0.0006103515625, 0.0054931640625, 0.07080078125, 0.68115234375, 5.2587890625]
```

(5) 初速 v0 を $10^{16}$ としたところ，以下の実行結果が得られた．この結果から，関数 numeric(g, v0, t, n) について，どのようなことが言えるだろうか．2 行程度で述べよ．丸め誤差，桁落ち，情報落ち，打切り誤差などとの関係があれば述べること．

```
> compare_v(10 ** 16)
[10.0, 10.0, 10.0, 10.0, 10.0, 10.0]
```

## 問題 3

配列 $a$ に $n$ 個の整数値が格納されているとする $(n \geq 2)$．$0 \leq i \leq n-1$ に対し，$a$ の中の $i$ 番目の要素を $a_i$ としたとき，$\sum_{i=j}^{k} a_i$ を $a$ の $j$ から $k$ までの部分和と呼ぶ（ただし $0 \leq j \leq k \leq n-1$）．全部で $\frac{n(n+1)}{2}$ 通りある $a$ の部分和のうち最大値を求めるアルゴリズムを考える．例えば，$a = [1, -9, 3, -1, 6]$ の場合には $3 + (-1) + 6 = 8$ が答となる．以下の問に答えよ．

(1) アルゴリズムの 1 つを以下に示す．空欄を埋めよ．

```
def func1(a, n):
  temp_max = a[0]
  for j in range(0, n):
    subsum = 0
    for k in [   A   ]:
      [   B   ]
      if [   C   ]:
        temp_max = subsum
  return temp_max
```

(2) 関数 `func1(a, n)` の計算量オーダを `n` を用いて表わせ．

(3) 配列 $a$ に丁度 1 つだけ負の値が含まれると仮定する．この仮定の下で効率の良いアルゴリズムを以下に示す．空欄を埋めよ．

```
def func2(a, n):
  subsum = 0
  for i in range(0, n):
    [   D   ]
    if a[i] < 0:
      temp_max = subsum - a[i]
      if [   E   ]:
        subsum = 0
  if [   F   ]:
    temp_max = subsum
  return temp_max
```

(4) 関数 `func2(a, n)` の計算量オーダを `n` を用いて表わせ．

(5) 問 (3) の仮定が満たされない場合にも効率の良いアルゴリズムを以下に示す．空欄を埋めよ．

```
def func3(a, n):
  temp_max = a[0]
  subsum = 0
  for i in range(0, n):
    [   G   ]
    if [   H   ]:
      temp_max = subsum
    if [   I   ]:
      subsum = 0
  return temp_max
```

(6) 関数 `func3(a, n)` の計算量オーダを `n` を用いて表わせ．