

## 問題 1

時間の長さを、時間、分、秒で表す Ruby のクラス **Time** が次のように定義されているとする。

```
class Time
  def initialize(h,m,s)
    @hour = h
    @minute = m
    @second = s
  end
  def to_string
    @hour.to_s + "時間" + @minute.to_s + "分" + @second.to_s + "秒"
  end
end
```

以下で、初期値が与えられた **Time** 型の値は、条件

$$0 \leq \text{@hour}, \quad 0 \leq \text{@minute} < 60, \quad 0 \leq \text{@second} < 60$$

が成立しているものと仮定してよい。しかし、メソッドを実行して状態が変化した **Time** 型の値は、上の条件を満たすようにする必要がある。

1. 秒単位のデータを引数として与えられ、その秒数だけ現在の時間に加えたクラス **Time** のメソッド、**advance** を定義しなさい。与えられるデータは 0 以上の整数と仮定してよい。メソッド定義だけ書けばよく、クラス定義を写す必要はない。
2. 秒単位のデータを引数として与えられ、その秒数だけ現在の時間から引いたクラス **Time** のメソッド、**back** を定義しなさい。与えられるデータは 0 以上の整数と仮定してよい。また、引いた結果の時間がマイナスになることはないと仮定してよい。メソッド定義だけ書けばよく、クラス定義を写す必要はない。

たとえば、次のように実行すると、

```
t = Time.new(5,3,43)
puts t.to_string
t.advance(10000)
puts t.to_string
t.back(20000)
puts t.to_string
```

次のような結果が得られる。

```
5 時間 3 分 43 秒
7 時間 50 分 23 秒
2 時間 17 分 3 秒
```

## 問題 2

配列 **a** の中からある値 **v** がしまわれている場所(添字)を探すことを考える。ただし、**a** は整数がしまわれている配列で、**v** は探す値(整数)であり、**a** の中には必ず **v** と同じ値が 1 つだけあるとする。

1. 先頭から順に配列の要素を調べ、最初に **v** と同じ値がしまわれている場所を見つけたときにその添字を返すようなアルゴリズムに基づいたメソッド **linear(a,v)** の定義を書け。

2. 配列 **a** が昇順(数が大きくなる順)に整列されているとする。このとき、次のようなアルゴリズムが考えられる。

(ア) 最初は調べる範囲を配列全体とする

(イ) 調べる範囲の中央の値と **v** を比べ、(**v** と等しくなかった場合には)大小関係を用いて、調べる範囲を前半分あるいは後半分に絞り込むことを続ける

このアルゴリズムに基づいたメソッド **binary(a,v)** の定義を書け。

3. 1 と 2 で定義したアルゴリズムの計算量を求めよ。

※ 参考のために「正の数から成る配列 **a** の中で、先頭から最初に出現する 0 までの間(0 がなければ全範囲)の最大値を見つけて返す」Ruby のメソッド **max(a)** の定義例を以下に示す。

```
def max(a)
  v = 0
  i = 0
  while i < a.size && a[i]!=0
    if v <= a[i]
      v = a[i]
      i = i + 1
    else
      i = i + 1
    end
  end
  v
end
```

### 問題 3

1. 計算機で数値計算のプログラムを書くときには様々な誤差に気をつけなくてはならないが、以下の各々の誤差の定義とその誤差が現れる例を簡潔に説明せよ。

(ア) 丸め誤差

(イ) 情報落ち誤差

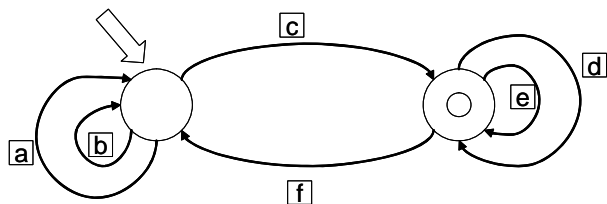
(ウ) 打ち切り誤差

(エ) 桁落ち誤差

2. 常微分方程式の数値解法には Euler 法と RungeKutta 法があるが、それぞれの特徴を誤差の観点から簡潔に述べよ。

### 問題 4

3 進数をアルファベット  $\Sigma = \{0,1,2\}$  上の文字列によって表現するとき、奇数を表す文字列を受理するオートマトンを以下の図のように定義する。a から f までの四角に入る文字をひとつずつ書け。



以上