

問題 1

1.

```
def advance(s)
  @second = @second + s
  while @second >= 60
    @minute = @minute + 1
    @second = @second - 60
  end
  while @minute >= 60
    @hour = @hour + 1
    @minute = @minute - 60
  end
end
```

2.

```
def back(s)
  @second = @second - s
  while @second < 0
    @minute = @minute - 1
    @second = @second + 60
  end
  while @minute < 0
    @hour = @hour - 1
    @minute = @minute + 60
  end
end
```

解説

やることは普通の関数を書くときとほぼ一緒。

違うのは変数の取り扱いくらいです。@つけてやるとインスタンス関数となります。

問題 2

1.

```
def linear(a,v)
  i = 0
  while a[i] != v
    i = i + 1
  end
  i
end
```

2.

```
def binary(a,v)
  l = 0
  h = a.length()-1
  i = (l + h) / 2
  while v != a[i]
    if v < a[i]
      h = i - 1
      i = (l + h) / 2
    else
      l = i + 1
      i = (l + h) / 2
    end
  end
  i
end
```

3.

$\text{linear} \cdots \text{a.length()} / \text{binary} \cdots \log_2 \text{a.length()} + 1$ (底 2、真数 a.length())

解説

1,2 はそのまま。

3 の `linear` も分かると思います。`binary` の `+1` は `a.length()` が 2 の累乗 `+1` の時。正確にするならガウス記号でくくってやるべきだけどそこまで細かくしないでいいと思います。

問題 3

(ア)10 進法だと正確に表せるが 2 進法だと有限桁では近似値になってしまうことにより生じる誤差。ex) $1/10(0.1(10) = 0.000110011\dots(2))$

(イ)大きな数に小さな数を足した時、小さな数が大きな数の有効桁の範囲外になることで無視されてしまうことにより生じる誤差。ex) $10^{**8} + 10^{**(-4)} = 10^{**8}$

(ウ)無限級数によって表されている値を、適当な数の項まで計算することで近似値を求める時、以降の計算を打ち切ることにより生じる誤差。ex) Taylor 展開

(エ)非常に近い二つの値の差を取った時にその差の有効桁数が低くなることにより生じる誤差。ex) $(2.00001)^{**2} - 2^{**2} = 0.0004$

解説

(イ)(エ)は有効数字の問題と同じように考えるとすんなり分かると思います。

(ウ)も Taylor 展開をどこまでやるかで値の正確さが違うだろうと言うことは分かると思います。

(ア)だけは数学やって出てくることはまず無い問題なので他より理解しづらいかも知れませんが、読めば分かる程度ではないかと。

問題 4

a.0 b.2 c.1 d.0 e.2 f.1 (a と b、d と e は順不同)

解説

オートマトンの問題です。うちのクラスはやんなかったけど、クラスによっては課題でやったみたいだね。俺も他のクラスの課題をやらされたやらせてもらったことがあります。(一応言っとくと俺は教えただけでまる写しさせたわけじゃないです。念のため。) この話はこれ以上は愚痴になるからいいや。

さて、オートマトンですが、簡単に説明すると、真ん中の大きい○の間を行ったり来たりします。スタートは大きな矢印がついてる○(つまり左)。そこからいずれかの数字が入力されたら、その数字に対応する細い矢印を辿ります。最後に◎の方(つまり右)で止まっていれば OK。ちなみにここでアルファベット 123 とか書いてますが、単に文字だって言ってるだけです。所謂アルファベット(AtoZ)ではないです。

次に 3 進数の説明ですが(要らんとするけど)、例えば 1020201(3)は 10 進数で $1*(3^{**6}) + 0*(3^{**5}) + 2*(3^{**4}) + 0*(3^{**3}) + 2*(3^{**2}) + 0*(3^{**1}) + 1*(3^{**0})$

これの奇偶を調べるわけですが、気付いたと思うけど各項は $(0\sim 2) \cdot (\text{奇数})$ の形になっているわけですね。で、奇数ってのは奇数を奇数個集めた和なので、各項のうち奇数項が奇数個あればいいわけです。なので奇数が出たら奇数→偶数、偶数→奇数と移動すればいいのです。

さて、長く書いたけど内容は非常に単純です。オートマトンがどういう動きするのか理解出来れば難しい所はありません。オートマトンが分からんって人は直接聞いて下さい。文章で説明するのが難しいし、図を用意するのが面倒だし。