4. Tossing a coin

a.) We toss the coin 2 times and there are only 4 situations that can happen:

① the first toss is head, the second toss is tail
② the first toss is tail, the second toss is head
③ Both toss are heads
④ Both toss are tails.

The probability of ① happens is: $\frac{2}{3} \times \frac{1}{3} = \frac{2}{9}$
The probability of ② happens is: $\frac{1}{3} \times \frac{2}{3} = \frac{2}{9}$
The probabity of ③ happens is: $\frac{2}{3} \times \frac{2}{3} = \frac{4}{9}$
The probabity of ④ happens is $\frac{1}{3} \times \frac{1}{3} = \frac{1}{9}$

we can see that ①② happens with equal probability.
As a result, we can toss a coin 2 times, if it ~~outpt~~ outputs (head, taill) we output 1, if it output (~~head~~ tail, head), we output 0. For cases that output (0,0), (1,1) we do the experiment again. Here is the persudo code:

```
a = toss_unfair_coin
b = toss_unfair_coin
if [a == b == head] or [a == b == tail]:
  repeat.
if [a = head] and [b = tail]:
   output 1
if [a = tail] and [b = head]:
   output 0
```

no. expected experiment

The average number of coin flips to output a number is: $\frac{9}{4} \times 2 = 4.5$

number of tosses in each experiment

b) Following the same reasoning from the first part:

The probability of [head, tail] in 2 flips is:

$p(1-p)$

which is the same as the probability of [tail, head] in 2 flips:

$(1-p)\cdot p$

The general picture of the algorithm is as follows:

a = toss - unfair - coin

b = toss - unfair - coin

if [a == head and b == head] or [a == ~~head~~ tail and b == tail]:

repeat

if [a = head] and [b = tail]: (with probability of $p(1-p)$

output 1.

if [a = tail] and [b = head]: (with probability of $(1-p)\cdot p$

The expected number of experiments to get [HT or TH]:

$$\frac{1}{p(1-p)+(1-p)\cdot p} = \frac{1}{2\cdot p(1-p)}$$

Each experiment requires to flip the coin twice:

$$\frac{1}{2(1-p)\cdot p} \times 2 = \frac{1}{(1-p)\cdot p}$$

Which is the avg. of coin fips.