cmput 204 **seminar 1** **fall 2015**

1. Using the longhand algorithm, find the largest integer x such that $x^2 \leq 69\ 00\ 70$

2. Using the longhand algorithm, find the square root of 835.8 correct to the 10^{-1} digit.

3. Using the longhand algorithm, and working in binary, find the largest integer x such that $x^2 \leq 11\ 01\ 10\ 11\ 11_2$.

4. `0b110 1010 0101 0111 1101 = 435581`.

   What is `0b1 1010 1001 0101 1111` in decimal?

   What is `871163` in binary?

5. On the assignment, your number $k = 11 + (n^{3191751} \pmod{89})$, where n is your student number. Write your student number and your number k. Check your answer, e.g. below.

```
def mynum(n):
  return pow(n,3191751,89)
```

6.
```
def foo(n):
  if n==0: return 0
  return 1 + n + foo(n-1)
```

   Assume that `foo(15)` returns $15 * 18/2$.

   Prove that `foo(16)` returns $16 * 19/2$.

7. fib(n) $\approx 1.618^n$. So, roughly how many bits are in (the binary representation of) fib(n)? And how many bits are in fib(2n)?

8. On my laptop, executing `fib(36)` takes about 7.4 seconds. Estimate the number of years it would take to execute `fib(100)`. (Ignore the fact that my laptop would not have enough memory for this computation.) Justify carefully.

1.
```
                 8   3   0
               --------
            \/ 69  00  70
8              64
               --
                5  00
163             4  89
                ----
                  11  70
1660              00  00
                  -----
                  11  70
so 690070 = 830 squared plus 1170.
```

2.
```
               2   8. 9
             ---------
            \/ 8  35.80
2              4
             --
               4  35.
48             3  84.
               ----
                 51.80
569              51.21
                 -----
                   .59
so 835.8 = 28.9 squared plus 0.59
```

3.
```
                 1   1   1   0   1
               ---------------
            \/ 11  01  10  11  11
1              1
               --
               10  01
101             1  01
                ----
                1  00  10
1101               11  01
                -------
                   1  01  11
11100              0  00  00
                ----------
                   1  01  11  11
111001                11  10  01
                      --------
                      10  01  10
so 11 0110 1111 = 1 1101 squared plus 10 0110
```

4. Let x, y be the two binary numbers, and a, b the two decimal numbers. So $x = a$.

   Notice that x consists of the same bitstring as y, followed by `01`. So $x = 4y + 1$. So $y = (x-1)/4 = (a-1)/4 = 435580/4 = 108970$.

   Notice that $y = 2x + 1$. So `y = 2a+1`

   ```
   = 2*(0b110 1010 0111 1101) +1
   =   0b1101 0100 1111 1010 +1
   =   0b1101 0100 1111 1011 .
   ```

6. `foo(n)` returns `1+n+foo(n-1)`, so `foo(16)` returns `1+16+15*18/2 = 152 = 8*19 = 16*19/2` .

7. The number of bits in the binary representation of the positive integer n is $1 + \lfloor \lg n \rfloor$, so the number of bits in `fib(n)` is roughly $1 + \lg(1.618^n) = 1 + n \lg 1.618 = 1 + .694n$.

   The number of bits in `fib(2n)` is roughly $1+\lg(1.618^{2n}) = 1+2n \lg 1.618 = 1+1.388n$.

8. The runtime is in $\Theta(\alpha^n)$ where $\alpha \approx 1.618$, so the runtime of `fib(100)` is about α^{100-36} times the runtime of `fib(36)`, so about

   $7.4\alpha^{64}/(60 * 60 * 24 * 365.25) \approx 5.56e6$ years.