README for Bignum

Vishal Bindal
2018CS50425

Bignum is implemented as a struct, with the following members:

- char* array: to store digits in the bignum as a string
- int len: length of the above string
- int dec: position of decimal in the number, relative to right of the last digit in the string
- char sign: '+' or '-'

Operator implementations:

- ADD:
  - First equalise the decimal parts of the 2 bignums
  - Then add the 2 numbers digit by digit, maintaining a carry
  - If one of the numbers is negative, subtract and then adjust the sign. If both are negative, then add and adjust the sign.
- SUBTRACT:
  - First equalise the decimal parts of the 2 bignums
  - Find Tens' complement of the second no (which is to be subtracted from the first)
  - Then add it to the first no
  - If there's a carry, then the final result is positive, discard the carry and shift the rest
  - Else result is negative. Take tens complement and set sign to negative
- MULTIPLY
  - Set sign of product as '+' if signs of bn1 and bn2 are same, '-' otherwise
  - Multiply bn1 by each digit of bn2 (by multiplying each digit of bn1, maintaining a carry)
  - Set decimal position of product using decimal positions of bn1 and bn2
- DIVIDE
  - Ensure length of dividend to be atleast same as divisor
  - Take a big number bn with first k digits from dividend, where k is length of divisor.
  - In each step, find max factor mf such that divisor*mf < bn, then do bn= bn - divisor*mf. Then increase length of bn by 1 and carry next digit from dividend
  - Continue till 20 decimal places reached
- SQROOT
  - Ensure 40 places in decimal to maintain 20 decimals in answer
  - Similar to divide, just calculate the factor like (divisor@mf)*mf < bn. Then set divisor = divisor@mf + mf. Carry 2 digits from dividend at a time
- ABS
  - If sign is '-', then set to '+'.

    - Trim trailing and leading zeros
- POW
    - Use the algorithm pow(a,b) = if b is even then pow(a*a, b/2) else a*pow(a, b-1)
    - If -ve power is given, then take reciprocal of base and then calculate power.