Data Representation I

Olga Bystrova

**Exercise**: build a list of all four-digit binary numbers, and compute the corresponding single-digit hexadecimal number.

**BINARY ➔ HEXADECIMAL**

0001 ➔ 1

0010 ➔ 2

0011 ➔ 3

0100 ➔ 4

0101 ➔ 5

0110 ➔ 6

0111 ➔ 7

1000 ➔ 8

1001 ➔ 9

1010 ➔ A

1011 ➔ B

1100 ➔ C

1101 ➔ D

1110 ➔ E

1111 ➔ F

**Exercise**: Pretend we use a naïve floating-point format with 5bit mantissa and 3bit exponent (base-2). What is the smallest possible positive number representable? What is the largest positive number representable? The first bit of each is used for sign:

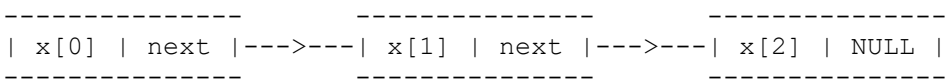
```
        [±****][±**]
         ^^^^^  ^^^--- exponent
           |---------- mantissa

Smallest Positive: [00001][111] (base-2: 0.0001)
Largest Positive: [01111][011]
```

**Exercise**: what is the best approximation of 0.01?

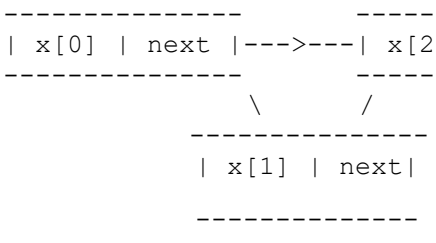
```
Decimal('0.01000000000000000020816681711721685132943093776702880859375')
```

**Exercise**: use diagrams like the above to explain how to delete an item from a linked list.

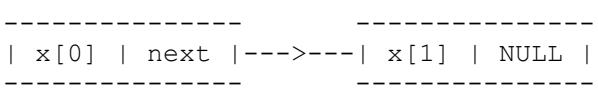
Beginning:




Process:

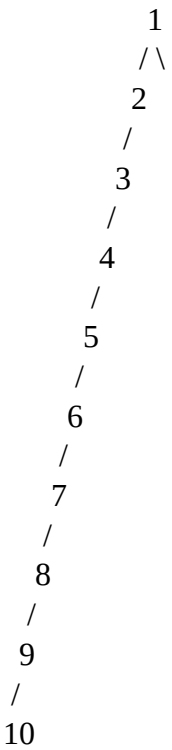



Finally:




**Exercise**: assemble the numbers 1-10 into binary search trees which are (a) maximally unbalanced to the left, (b) balanced, (c) one step from balanced.

(a)



(b)