Tutorial 1 - C and Data Representation — Week 3

1. Variable Range Limit & Data Representation

(a). $2\,147\,483\,647 = 2^{31} - 1$

↳ Largest value an int (32-bit) can take

$$\overbrace{0111\ldots1111}^{31\ 1\text{'s}}$$

(b). Output: Start, i is 0x7fffffff

↳ Basically the hexadecimal representation of i

$$\overbrace{1000\ldots0000}^{31\ 0\text{'s}}$$

(c). Output: What ?! i is 0x80000000

↳ Overflow has occurred (i is represented as a negative integer)

(d). 0x80000000 representation:

S&M : -0

1s : $-2\,147\,483\,647$ $(2^{31}-1)$

2s : $-2\,147\,483\,648$ (2^{31})

Excess : unlikely

2s ? Not sure why though

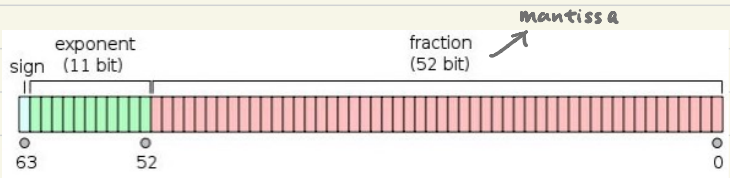
(e). Java:

```
jshell> int i = 2147483647
i ==> 2147483647

jshell> i + 1
$2 ==> -2147483648

jshell> String.format("Hex: %#X", i)
$3 ==> "Hex: 0X7FFFFFFF"

jshell> String.format("Hex: %#X", i + 1)
$4 ==> "Hex: 0X80000000"
```

Clearly using 2s-complement

2. Floating Point Representation

(a).



(b). Output : `0.1 is represented as 0x3fb999999999999a`

sign : 0x0

exponent : 0x 3fb $= (1019)_{10}$

mantissa : 0x b999...9a $=$ (52241755677497754)10

(12)

According to Google, representation of exp : excess - 1023

(c).