Algorithm 1

Divij Singh

31/01/18

1 Polynomials

1.1

def poly(num):
$P_n(x) = poly(num - 1) + a_n * x^{n-1} * x$

1.2

a.
$1^5 1^4 0^3 0^2 1^1 1^0$
$2^5 + 2^4 + 0 + 0 + 2 + 1$
$32 + 16 + 3$
51
Ans: 51_{10}

b.
$2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$
$256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$
511
Ans:511_{10}

c.
$2^3 + 0 + 0 + 1 + (1/2) + 0 + 0 + (1/2^4)$
$9 + 0.5 + 0.0625$
9.5625
Ans:9.5625_{10}

d.
$1.1111... = 1 + (1/2) + (1/2^2) + (1/2^3) + (1/2^4)...$
$1 + (0.5/(1 - (1/2)))$
$1 + (0.5/0.5)$
2

Ans:2_{10}

1.3

2 2s Complement

2.1

I will do this by taking the first bit of the number, and stating that if it is 1, then the number is negative, and if it is 0, the number is positive. If all bits after the first are 0, then the number is the one after 11111111, i.e $10000000_2 = -128_{10}$

2.2

a. Ans:01100100
b. Ans:01111111
c. Ans:11011101
d. Ans:10000000

2.3

The largest number is $2^n - 1$, and the smallest is -2^n

3 Floating Point

3.1 a

.
$20.17_{10} = 10100.001010111000001_2$
$1.010000101011100001 * 2^4$
$3 + 4 = 7_{10} = 111_2$
Ans:0|111|010000101011100001

b.
$10.3_{10} = 1010.010011001_2(1001 repeating)$
$1.010010011001 * 2^3$
$3 + 3 = 6_{10} = 110_2$
Ans:0|110|010010011001

3.2

1. Append $*2^0$ at the end of the binary number.
2. Move the decimal till there is one bit to the left of the point. Adjust the exponent of two so that it matched the number of places moved.
3. Keep all the numbers to the right of the decimal aside as the Mantissa.
4. Add the bias to the value of exponent of two. For 32 bit, it is 8. The bias is found using $2^(k-1)-1$ where k is the number of bits.
5. Set sign bit as 0 for positive, 1 for negative.
6. Write as (signed bit)|(bias plus exponent)|(mantissa)