Assignment 2 – Week 3.

Floating point numbers.

1: 8-bit floating-point representation of your number.

Student number: s3855159

A = 9, B = 5.

X = A + B/10 = 9 + 5/10 = 9.5.

Is odd so negative: X = -9.5_10

9.5_10 = 1001.1_2

Move decimal point three spaces left: 1.0011

Mantissa = 0011.

N = 3.

Bias = 2^N-1 – 1 = 2^3-1 – 1 = 2^2 – 1 = 4-1 = 3.

Exponent = N + bias = 3 + 3 = 6.

Add sign bit (1 for negative) + exponent + mantissa = **1110.0011**

2: The range of your 8-bit expression.

Exponent bits = 3.

Range = 0 to 7 or 2^-3 to 2^+4

Or

Maximum positive value = 7.9375

3: The precision (accuracy) of your 8-bit expression.

Precision is 4 bits.

4: Did you lose accuracy/precision through this 8-bit representation? What is causing this loss of precision?

I did not, because my number was equivalent to a half and could be rounded perfectly. I did other practice calculations with other numbers though and found that the loss of precision occurs very quickly with such a small number of bits allocated to the mantissa. With more bits it improves, but it is never truly precise. Just very close.