

IC220: HW 5b

Due: 18 Mar 2019

Full Name: _____ **Alpha:** _____

Circle Your Section: Aviv/1001 Aviv/2001 Aviv/4001 Choi/5001 Missler/5002

Total Points: 50

Preliminary: Carefully do the assigned reading for Chapter 2 (2.1-2.3,2.5-2.10,2.12)

1. Convert the following 32-bit, single precision float into a decimal (base 10) number. You can leave your answer in reduced fraction form or in decimal. For convenience, the number is broken with hyphens for different segments of the encoding. *(Note, you can use a calculator for this, but you'd be expected to do this by hand, without a calculator, on a exam.)*

(a) [5 points]

1 - 0 1 1 1 1 1 0 0 - 1 0

(b) [5 points]

0 - 1 0 0 0 0 0 1 0 - 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

- [5 points] Show the encoding of 12.6875 in 32-bit, single precision float. (*Note, you can use a calculator for this, but you'd be expected to do this by hand, without a calculator, on a exam.*)
- [5 points] Show the encoding of -7.25 in 32-bit, single precision float. (*Note, you can use a calculator for this, but you'd be expected to do this by hand, without a calculator, on a exam.*)
- [5 points] Show the encoding of -0.125 in 32-bit, single precision float. (*Note, you can use a calculator for this, but you'd be expected to do this by hand, without a calculator, on a exam.*)

5. [5 points] Convert the following C code to MIPS. You can assume single precision floats, and use pseudo instruction `li.s`.

```
float pick (float G[], int index){  
    return G[index];  
}
```

6. [5 points] Convert the following C code to MIPS. You can assume single precision floats, and use pseudo instruction `li.s`.

```
float maxdiv(float A, float B){  
    if(A > B) return A/B;  
    else      return B/A;  
}
```

7. [5 points] Convert the following C code to MIPS. You can assume single precision floats, and use pseudo instruction `li.s`.

```
float sum(float A[], int N){
    int j;
    float sum = 0.0;
    for (j=0; j<N; j++){
        sum = sum + A[j];
    }
    return sum;
}
```

8. [5 points] Convert the following C code to MIPS. You can assume single precision floats, and use pseudo instruction `li.s`.

```
float foo(float x, float y){
    if (x > y)
        return x + y;
    else
        return x - y;
}
```

9. [5 points] Convert the following C code to MIPS. Note: use **integers** not floats here! Also, use **mult** instruction that we learned in class that takes just 2 arguments.

```
int muskrat(int g, int h){
    int prod = g * h;
    if (prod < 0)
        prod *= -1;
    return prod;
}
```