Question 2 (20 points): When building models for deep learning, lower-precision floating-point representation can be used. Thus, there is a growing interest in floating-point representation that use fewer bits. Last year a minifloat representation was proposed. Minifloat borrows the same ideas used in floating-point formats to represent integer values in a large range using only 8 bits. An 8-bit minifloat has 1 sign bit, 4 exponent bits and 3 mantissa bits and an exponent bias equal +2 for normalized numbers.

- Minifloat has representations for +infinity (`0 1111 000`) and -infinity (`1 1111 000`).
- Minifloat has a representation for Not-a-Number (NaN): `x 1111 yyy` where `yyy` ! = `000`.
- When the exponent is `0000` the number represented is subnormal, and the value is `0.mmm` $\times 2^3$ where `mmm` are the three bits of the mantissa.
- When the exponent is not zero, then the number represented is normalized. Given a representation `s eeee mmm`, the value represented is `1.mmm` $\times 2^{e+2}$ where e is the value of the exponent `eeee` in the representation.
- The standard rules of rounding to the nearest even apply to minifloat.

1. (**5 points**) Which is the smallest, non-zero, positive value that can be represented in minifloat? Provide the binary representation and the decimal value.

   The smallest non-zero positive value that can be represented is:
   `0 0000 001` which is in denormalized form. The value is given by: $0.001 \times 2^3 = 1$

2. (**5 points**) Which is the largest positive value that can be represented in minifloat? Provide the binary representation and the decimal value.

   The largest positive value that can be represented is:
   `0 1110 111` which is in normalized form.

   The value is given by:
   $1.111 \times 2^{14+2} = 1\ 1110\ 0000\ 0000\ 0000 = 2^{17} - 2^{13} = (2^4 - 1) \times 2^{13} = 15 \times 8 \times 1024 = 122,880$

3. (**5 points**) There are many numbers within the range between the smallest and the largest value that cannot be represented precisely in minifloat. In such a case the value is rounded to the nearest value. What is the binary representation of 46_{10} in minifloat? Which is the actual value of the number represented after rounding?

   $46 = 101110 = 1.01110 \times 2^5 = 1.01110 \times 2^{3+2}$
   Rounding up to the nearest even: $1.100 \times 2^{3+2}$
   Binary representation: `0 0011 100` = `0001 1100` = `0x1C`
   The value represented is $1.100 \times 2^5 = 110000 = 32 + 16 = 48$

4. (**5 points**) A hardware that supports minifloat has an adder with a guard bit, a round bit, and a sticky bit. Show how the operation 5+64 is performed in this hardware, and what is the value of the result both in binary and in decimal. What is the minifloat representation of the result?

   $5 = 101 = 1.01 \times 2^2$
   $64 = 1000000 = 1.000 \times 2^6$
   We must adjust 5 to make the exponents the same:
   $5 = 0.000101 \times 2^6$

```
           G R S
 5 = 0.000 1 0 1
64 = 1.000 0 0 0
----------------------
       1.000 1 0 1
```

   Must round up, thus the result is: $1.001 \times 2^6 = 1001000 = 2^6 + 2^3 = 64 + 8 = 72$
   For the minifloat representation, $1.001 \times 2^6 = 1.001 \times 2^{4+2}$, thus:
   minifloat = `0 0100 001`