1. (10 points) Consider the floating point number system consisting of all numbers $\pm d_1 d_2 \times \beta^e$, where $\beta = 3$, $-1 \leq e \leq 1$, $0 < d_1 < 3$, and $0 \leq d_2 < 3$. Write down in rational form all of the positive numbers in this system. What is the largest absolute error incurred in adding two positive numbers in this system followed by rounding to the nearest number in this system, assuming there is no overflow, i.e., that the sum is not greater than the largest number in the system?

2. (10 points) In a floating point number system, there is an important number called machine epsilon (a.k.a. unit roundoff, unit rounding error, etc.). This is defined to be the smallest floating point number which, when added to one, results in a floating point number greater than one. It depends on the precision used in the floating point number system and varies when different precisions are used.

Implement and run the following algorithm for approximating machine epsilon.

```matlab
macheps = 1;
while 1 + macheps > 1,
    macheps = 2 * macheps;
end
macheps = 2 * macheps
```

Notes: This is a complete MATLAB script, i.e., it will run and produce the desired result in MATLAB with no additions or modifications. You may use an alternative to MATLAB for this assignment, but I encourage you to use MATLAB in order to begin familiarizing yourself with it.

You can enter this script in MATLAB by typing “edit” in the command window and then typing the script into the edit window. To run the script, type (command)-return on a Mac or (control)-return in Linux and Windows.

In MATLAB, the last line does not end in a semicolon and consequently will result in printing out `macheps`.

If you use MATLAB for this assignment, then, at the command prompt, type “eps” and compare the result to the output of your algorithm.