

## Recitation Assignment #2

Please do the following problems during your recitation session, including any additional problems given to you by your TA. Within 72hrs of your recitation session, you must upload the complete solutions to these problems to Sakai, so that your TAs can evaluate them in a timely fashion. **Please submit either a any .m files to Sakai. Your .m files should be named in the following format: NETIDRecitation2Problem#, where NETID is your NetID, and # is the problem number. For instance if your NetID was aaa111, and you were answering problem 3, your .m file would be named aaa111Recitation2Problem3.m.** Collaborative problems can be worked on in teams of up to 5 people, as long as each team member individually completes the problem uploads the solution as part of their own Sakai submission or shows the solution to their instructor individually, and lists the names of all collaborators in the Sakai submission. Collaboration and discussion of solutions is not permitted for questions labeled as individual problems.

1. [Collaborative] Engineers frequently need to convert measured quantities to a different scale. An example is temperature. A temperature can be converted from the Fahrenheit scale to the Celsius scale using the formula

$$c = \frac{5}{9}(f - 32)$$

Where  $f$  is the temperature in Fahrenheit and  $c$  is the resulting temperature in Celsius. Write a script to prompt the user for the temperature in Fahrenheit, and then display the temperature in Celsius as follows:

```
The temperature in degrees Celsius is:  
10
```

Here “10” should be replaced by the result of your calculation. **Please follow file naming instructions at the beginning of this document.**

2. [Collaborative] Many engineering applications such as the Global Positioning System need to describe and calculation locations or distances on Earth. This is complicated by the complex shape of the Earth. While a simple approximation of the Earth is a sphere, a more accurate approximation is a shape called an *oblate spheroid*. This means that the earth’s radius is larger along the equator than it is at the poles by about 20km! Think of a beach ball that you are squeezing slightly at the top and bottom.

To observe the difference between a spherical approximation and an oblate spheroid approximation, we will calculate the surface of the earth for both.

The surface area of an oblate spheroid is given as:

$$A(r_1, r_2) = 2\pi(r_1^2 + \frac{r_2^2}{\sin(\gamma)} \log(\frac{\cos(\gamma)}{1-\sin(\gamma)}))$$

Where  $r_1$  is the equatorial radius,  $r_2$  is the polar radius, and

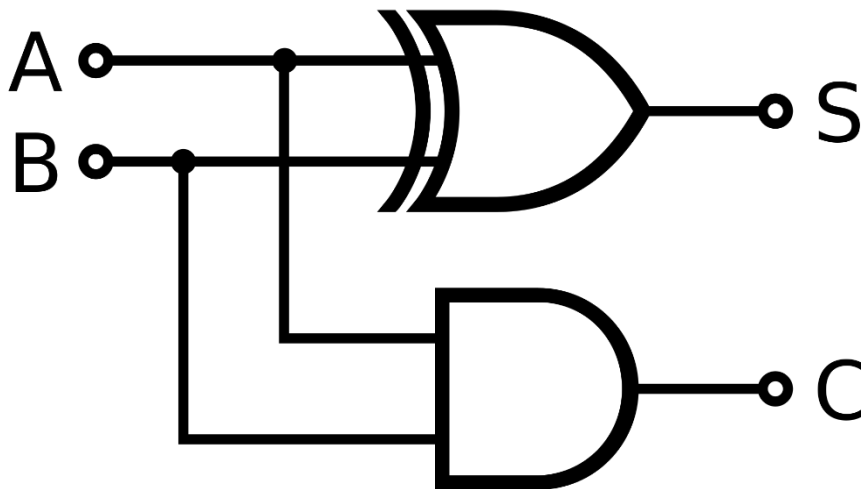
$$\gamma = \arccos\left(\frac{r_2}{r_1}\right)$$

We assume that  $r_2 > r_1$ . Write a script that asks the user to input the equatorial and polar radii and displays both the result of the equation above and the equation for a spherical approximation given as:

$$4\pi((r_1 + r_2)/2)^2$$

Along with a descriptive message. Apply the script to Earth's two radii, 6378.137km for equatorial ( $r_1$ ) and 6356.752km for polar ( $r_2$ ). **Please follow file naming instructions at the beginning of this document.**

3. [Individual] Recall how logic operations work, and the logic tables given in recitation and lecture in the last week. Also recall that, in computers, all mathematical operations are just combinations of mathematical operations. This figure is a logic diagram of a “half adder,” a way of performing binary addition on two bits:



A and B are the bits you are adding together. S is the resulting bit, and C is the carry value. Using MATLAB's logical operators, create a script that simulates this half adder. It should produce two Boolean variables that represent S and C on the diagram. Comment your code. **Please follow file naming instructions at the beginning of this document.**

4. [Individual] Create a script that asks the user for how many bits they have in a binary word. Return the number of possible values a word of that length can have as a displayed number in the command line. Comment your code. **Please follow file naming instructions at the beginning of this document.**
5. [Individual] Random numbers can be very useful to engineers in some situations. In this exercise we will introduce you to MATLAB's random number tools. Later in this class, we will show you some examples of how random numbers are useful to engineers. Use the help documentation to read about the functions rand, randn, and randi. Create a script that creates three random variables:

- A random decimal number drawn from a uniform distribution between 5 and 10.
- A random decimal number from a normal distribution with mean 0.
- A random integer that simulates a dice roll (that is, between 1 and 6).

Comment your code, and use good style when naming your variables. **Please follow file naming instructions at the beginning of this document.**