SM-2302 Labs (R1):

1. ```r
   f <- function(x) {
     # Check small prime
     if (x > 10 || x < -10) {
       stop("Input too big")
     } else if (x %in% c(2, 3, 5, 7)) {
       cat("Input is prime!\n")
     } else if (x %% 2 == 0) {
       cat("Input is even!\n")
     } else if (x %% 2 == 1) {
       cat("Input is odd!\n")
     }
   }
   ```

   Without running the above code, what do you expect the outcome will be for each of the following? Verify your answers by executing these functions in R.

   a. `f(1)`
   b. `f(3)`
   c. `f(8)`
   d. `f(-1)`
   e. `f(-3)`
   f. `f(1:2)`
   g. `f("0")`
   h. `f("3")`
   i. `f("zero")`

2. What is the output of the following code? Explain why.

   ```r
   z <- 0
   f <- function(x, y, z) {
     z <- x + y
     g <- function(m = x, n = y) {
       m / z + n / z
     }
     z * g()
   }

   f(1, 2, x = 3)
   ```

3. Write a set of conditional(s) that satisfies the following requirements,

   - If `x` is greater than 3 and `y` is less than or equal to 3 then print "`Hello, World!`"
   - Otherwise if `x` is greater than 3 print "`!dlroW ,olleH`"
   - If `x` is less than or equal to 3 then print "`Something else ...`"
   - Stop execution if x is odd and y is even and report an error, don't print any of the text strings above.

   Test out your code by trying various values of `x` and `y` and show that it works. *Hint: It might be helpful if you sketched a flow diagram before coding your function.*

4. Below is a vector containing all prime numbers between 2 and 100:
   ```

```r
primes <- c( 2,  3,  5,  7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
            43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97)
```

If you were given the vector `x <- c(3, 4, 12, 19, 23, 51, 61, 63, 78)`, write the R code necessary to print only the values of `x` that are *not* prime (without using subsetting or the `%in%` operator).

Your code should use *nested* loops to iterate through the vector of primes and `x`.

5. In a room containing n individuals, what is the probability that at least two of them share the same birthday? The answer might seem surprising at first, but this can easily be solved with some probability theory. Assume that

   - there are only 365 days in a year;
   - each day is equally likely to be a birthday; and
   - birthdays are independent of each other.

   It's easier to calculate the probability of the event complement, i.e. the probability *no one shares a birthday*. This means that each birthday must be unique, and there are ${}^{365}P_n = \frac{365!}{(365-n)!}$ ways of choosing the birthdays, out of a total 365^n possibilities. Therefore,

$$\Pr(\text{No common birthdays}) = \frac{365!}{(365-n)!365^n}, \tag{1}$$

   and that

$$\Pr(\text{At least two shares a birthday}) = 1 - \frac{365!}{(365-n)!365^n}. \tag{2}$$

   Your task is to write an R function called `bday_fn()`, which takes input a numeric vector and return the birthday probability in (2). Once you have done that, run the following code:

```r
n <- 2:100
prob <- bday_fn(2:100)
plot(x = n, y = prob, type = "l")
```

   Roughly how many individuals is needed to have this probability be 50%? 99.9%?

   Hint: Use the `lfactorial()` function in R. Find out what this accomplishes in the help files. You will need to manipulate the equations above so you can use this function.