Homework 1 Solutions

Coding Exercise 1

```
# load packages
# install.packages("stringr") # just run this once
library(stringr)

# now compute the number of characters in each element of x
x <- c("economics", "econometrics", "ECON 8080")
str_length(x)
```

```
[1]  9 12  9
```

Coding Exercise 2

Approach 1:

```
sum_one_to_n_1 <- function(n) {
    n_seq <- seq(1, n)
    sum(n_seq)
}
```

Approach 2:

```
sum_one_to_n_2 <- function(n) {
    n*(n+1)/2
}
```

Approach 3:

```
sum_one_to_n_3 <- function(n) {
    out <- 0
    for (i in 1:n) {
        out <- out + i
    }
    out
}
```

Compare the results

```
sum_one_to_n_1(100)
```

```
[1] 5050
```

```
sum_one_to_n_2(100)
```

```
[1] 5050
```

```
sum_one_to_n_3(100)
```

```
[1] 5050
```

Coding Exercise 3

Part (a):

```
fibonacci <- function(n) {
    # handle n = 1 and n = 2 separately
    if (n == 1) {
        return(0)
    } else if (n == 2) {
        return(1)
    }

    # initialize the sequence
    fib_seq <- c(0,1)

    # loop to compute the sequence
    for (i in 3:n) {
        fib_seq[i] <- fib_seq[i-1] + fib_seq[i-2]
    }

    # return the nth element
    return(fib_seq[n])
}

fibonacci(5)
```

```
[1] 3
```

```
fibonacci(8)
```

```
[1] 13
```

Part (b):

```
alt_seq <- function(a, b, n) {
    # handle n = 1 and n = 2 separately
    if (n == 1) {
        return(a)
    } else if (n == 2) {
        return(b)
    }

    # initialize the sequence
    alt_seq <- c(a,b)

    # loop to compute the sequence
    for (i in 3:n) {
        alt_seq[i] <- alt_seq[i-1] + alt_seq[i-2]
    }

    # return the nth element
    return(alt_seq[n])
}
alt_seq(3,7,4)
```

```
[1] 17
```

Coding Exercise 4

Part (a)

```
is_prime <- function(x) {
    # handle the case of 1 separately
    # (not a prime number)
    if (x == 1) {
        return(FALSE)
    }
    for (i in 2:(x-1)) {
        if (x %% i == 0) {
            return(FALSE)
        }
    }
    return(TRUE)
}
is_prime(7)
```

```
[1] TRUE
```

```
is_prime(10)
```

```
[1] FALSE
```

Part (b)

```
prime <- function(n) {
    one_to_n <- 1:n
    is_prime_vec <- sapply(1:n, is_prime)
    one_to_n[is_prime_vec]
}
prime(10)
```

```
[1] 3 5 7
```

Coding Exercise 5

Part (a)

```
nrow(iris)
```

```
[1] 150
```

Part (b)

```
mean(iris$Sepal.Length)
```

```
[1] 5.843333
```

Part (c)

```
setosa_subset <- subset(iris, Species == "setosa")
mean(setosa_subset$Sepal.Width)
```

```
[1] 3.428
```

Part (d)

```
sorted_iris <- iris[order(iris$Petal.Length),]
sorted_iris[1:10,]
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
23          4.6         3.6          1.0         0.2  setosa
14          4.3         3.0          1.1         0.1  setosa
15          5.8         4.0          1.2         0.2  setosa
36          5.0         3.2          1.2         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
17          5.4         3.9          1.3         0.4  setosa
37          5.5         3.5          1.3         0.2  setosa
39          4.4         3.0          1.3         0.2  setosa
41          5.0         3.5          1.3         0.3  setosa
42          4.5         2.3          1.3         0.3  setosa
```

Coding Exercise 6

```
quadratic_solver <- function(a,b,c) {
    postive_root <- (-b + sqrt(b^2 - 4*a*c))/(2*a)
    negative_root <- (-b - sqrt(b^2 - 4*a*c))/(2*a)
    list(positive_root = postive_root, negative_root = negative_root)
}
quadratic_solver(1, 4, 3)
```

```
$positive_root
[1] -1

$negative_root
[1] -3
```