# Class 13, Homework Solution

**Question 1 (2 points). Create your own function to calculate x square minus y square. Use the function to calculate the result when x is -10 and y is 6.**

```
minus.square <- function(x, y) {
    z <- x^2 - y^2
    return(z)
}
minus.square(x=-10, y=6)
```

```
## [1] 64
```

**Question 2 (2 points). Create your own function to calculate the sum of several numbers. Use the function to calculate the sum of 1.1, 2.2, and 3.3, and the sum of a matrix with integers from 1 to 4, 2 rows and 2 columns.**

```
sum.fun <- function(x) {
  s <- 0
  for (i in 1:length(x)) {
      s <- s + x[i]
  }
  return(s)
}

sum.fun(c(1.1, 2.2, 3.3))
```

```
## [1] 6.6
```

```
sum.fun(matrix(1:4, 2, 2))
```

```
## [1] 10
```

**Question 3 (2 points). Improve the function in question 2 to make it work for numbers saved as character values. Use the improved function on a vector with these values: '1', '10', '100', '1000'.**

```
sum.fun2 <- function(x) {
  s <- 0
  if (typeof(x) == 'character') {
      x <- as.numeric(x)
  }
  for (i in 1:length(x)) {
      s <- s + x[i]
  }
  return(s)
}
```

```
chr <- c('1', '10', '100', '1000')
sum.fun2(chr)
```

```
## [1] 1111
```

**Question 4 (4 points).** Use the code provided below to generate a random time series. Create a function to conduct temporal smoothing (i.e. calculate the mean of several consecutive numbers). Use the function to calculate the mean of every 3 consecutive numbers and every 9 consecutive number. Plot the original time series and two smoothed time series you created in a single graph.

```
ts <- rnorm(n=100, mean=10, sd=2)
```

```
tsmooth <- function(ts, n) {
    out <- numeric(length(ts))
    m <- floor(n/2)
    for (i in 1:length(ts)) {
      start <- ifelse(i-m < 0, 0, i-m)
      to <- ifelse (i+m > length(ts), length(ts), i+m)
      out[i] <- mean(ts[start:to])
    }
    return(out)
}
tss1 <- tsmooth(ts, n=3)
tss2 <- tsmooth(ts, n=9)
plot(ts, type='l', col=1)
lines(tss1, col=2, lwd=2)
lines(tss2, col=4, lwd=2)
```