CSE 6242 | Assignment 1 | Ebeid ElSayed | eelsayed3@gatech.edu

# Q1 - Something interesting

# Variables scope

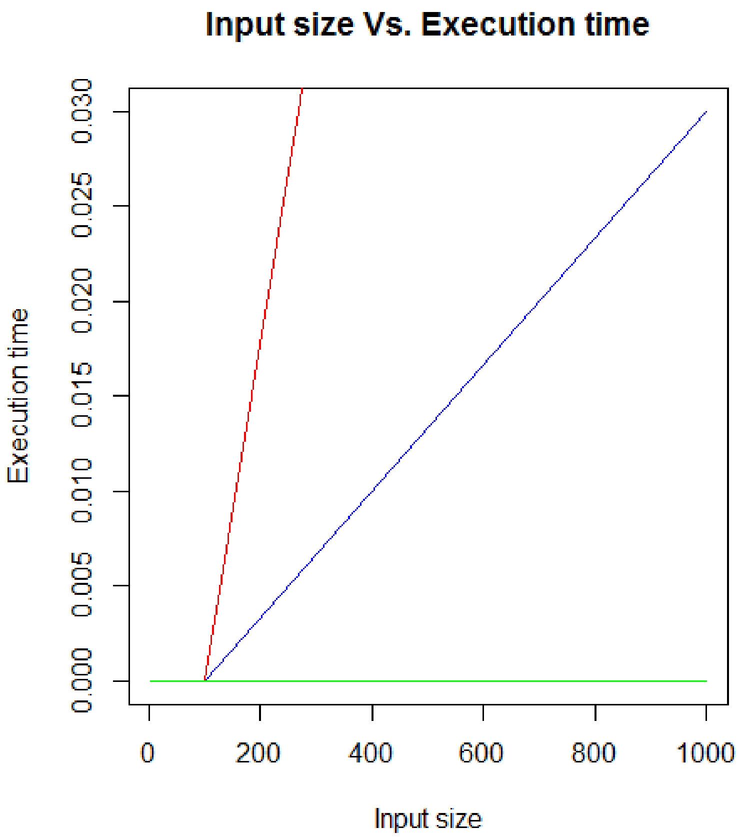
```
testFunction1 = function(x){
  x <- x + 1
  y <<- y +1
  return(x)
}
x <- 2
y <- 5
print(testFunction1(x))
print(x) # Didn't change inside the function
print(y) # Changed inside the function due to global access modifier usage <<-
```

# S3 classes

```
x <- c(1,2,3)
class(x) # "numeric"
class(x) <- append(class(x),"myClass")
class(x) # "numeric"  "myClass"

GetFirst <- function(x)
{
    UseMethod("GetFirst",x)
}

GetFirst.myClass <- function(x)
{
  return(x[1])
}

GetFirst(x) #1
```

```
# Q2 -  Log Gamma (Loop)
log_gamma_loop = function(x){
  x = x - 1;
  result = 0;
  while(x>0){
    result = result + log(x)
    x = x - 1;
  }
  return(result)
}
print(log_gamma_loop(5))

# Q3 - Log Gamma (Recursive)
log_gamma_recursive = function(x){
  if(x == 1) return(0)
  else return (log(x-1) + log_gamma_recursive(x-1))
}
print(log_gamma_recursive(5))

# Q4 - Sum of Log Gamma
sum_log_gamma_loop = function(x){
  sum = 0
  for(num in seq(1, x, by = 1)){
    sum = sum + log_gamma_loop(num)
  }

  return(sum)
}

sum_log_gamma_recursive = function(x){
  sum = 0
  for(num in seq(1, x, by = 1)){
    sum = sum + log_gamma_recursive(num)
  }

  return(sum)
}

sum_log_gamma_builtin = function(x){
  sum = 0
  for(num in seq(1, x, by = 1)){
    sum = sum + log(num)
  }

  return(sum)
}

# Q5 - Performance
if(!require(rbenchmark))
{
  message("installing the 'rbenchmark' package")
  install.packages("rbenchmark")
}
  df1 <- benchmark(sum_log_gamma_loop(5), sum_log_gamma_loop(10), sum_log_gamma_loop(100),
sum_log_gamma_loop(1000), order = NULL)
  df1["x"] <- c(5, 10, 100, 1000)
  df2 <- benchmark(sum_log_gamma_recursive(5), sum_log_gamma_recursive(10), sum_log_gamma_recursive(100),
sum_log_gamma_recursive(1000), order = NULL)
  df2["x"] <- c(5, 10, 100, 1000)
  df3 <- benchmark(sum_log_gamma_builtin(5), sum_log_gamma_builtin(10), sum_log_gamma_builtin(100),
sum_log_gamma_builtin(1000), order = NULL)
  df3["x"] <- c(5, 10, 100, 1000)

if(exists("df1"))
```

```
{
  plot(df1$x, df1$sys.self, xlab="Input size",ylab="Execution time", type = "l", col="blue", main="Input size Vs. Execution time")
  if(exists("df2"))
    lines(df2$x, df2$sys.self, type = "l", col="red")
  if(exists("df3"))
    lines(df3$x, df3$sys.self, type = "l", col="green")
}
```



Clearly the recursive implementation is the worst.