CSC 121H1 S 2018 Quiz 2 (Version B)
March 19, 2018
Duration — 35 minutes
Aids allowed: none

UTORid: _____

Last Name: _____ **First Name:** _____

Lecture Section: L0101 (MWF12)
Instructor: Mark Kazakevich

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This quiz is double-sided, and consists of 4 questions and a list of built-in R function descriptions. *When you receive the signal to start, please make sure that your copy is complete.*

- Read each question **carefully**.
- Comments are not required except where indicated, although they may help us mark your answers.
- No error checking is required: assume all user input and all argument values are valid.
- If you use any space for rough work, indicate clearly what you want marked.
- Do not remove any pages from the quiz booklet.
- You may use a pencil; however, work written in pencil will not be considered for remarking.
- In some cases, part marks will be awarded for partial answers.

1: _____/ 6

2: _____/ 4

3: _____/ 5

4: _____/ 5

TOTAL: _____/20

Total Pages = 8

Question 1. [6 marks]

The following commands are typed into the R console as shown below. (The > characters at the beginnings of the lines are the prompts printed by R, not part of what was typed.) In the six blank spaces below, fill in the output that R will produce as a result of these commands.
You do not have to add the [1] at the beginning of each output line.

```
> v <- c(6, 3)
> rep(v, 3)


> seq(0, 6, 1.5)



> M <- matrix(c(1, 2), nrow=2, ncol=2)
> M[1, 2] == 1



> L <- list(a=1:3, b=5:1, c=c("hey", "hi", "hello"))
> sum(L$a)




> L$c[L$b[3]]




> set.seed(5)
> runif(1)
[1] 0.2002145
> runif(1)
[1] 0.6852186
> runif(1)
[1] 0.9168758
> set.seed(5)
> rep(runif(1), 3)
```

Question 2. [4 marks]

Consider the following function, `ListMaker`, which returns a dictionary (also known as a named list).

```
ListMaker <- function(v) {

  L <- list()
  for (i in 1:length(v)) {
    if (i != length(v)) {
      valueCharacter <- as.character(v[i])
      if (valueCharacter %in% names(L)) {
        L[[valueCharacter]] <- append(L[[valueCharacter]], v[i + 1])
      } else {
        L[[valueCharacter]] <- v[i + 1]
      }
    }
  }

  return(L)

}
```

Consider the dictionary returned by calling the function on the following vector argument. For the function call below, list each of the keys of the returned dictionary, and the values for each key. Make sure you clearly indicate which of the keys or values are strings by placing "quotes" around them.

```
> ListMaker(c(1, 2, 2, 4))
```

Key	Value

Question 3. [5 marks]

Complete the function body of the function `EdgeReplace` according to its docstring and the test cases below. Do not use any functions we haven't talked about in class.

```
> Matrix1
     [,1] [,2] [,3]
[1,]    1    4    1
[2,]    4    4    1
[3,]    1    1    1
> EdgeReplace(Matrix1, 4, 1)
     [,1] [,2] [,3]
[1,]    0    4    0
[2,]    4    4    1
[3,]    1    1    1
```

```
> Matrix2
     [,1] [,2]
[1,]    3    3
[2,]    3    3
> EdgeReplace(Matrix2, 3, 1)
     [,1] [,2]
[1,]    3    3
[2,]    3    3
> EdgeReplace(Matrix2, 3, 2)
     [,1] [,2]
[1,]    0    0
[2,]    0    0
```

```
EdgeReplace <- function(M, n, times) {
  # Returns a copy of numeric matrix 'M' which is changed in the following way:
  # For any row that contains exactly 'times' number of elements equal to 'n',
  # that row has its first and last elements replaced with 0's.
  #
  # Precondition: nrow(M) >= 1, ncol(M) >= 2
```

```
}
```

Question 4. [5 MARKS]

The following are the contents of a file called `account.txt`

```
transactionID amount balance
1 -10 90
2 -30 60
3 10 70
4 -20 50
```

Each row of `account.txt` (except for the header at the top) represents the transactions in a bank account that starts with 100 dollars, including the transaction number (which always starts at 1 and increases by 1), the amount of the transaction (positive for a deposit, negative for a withdrawal), and the amount of money in the account after the transaction (the balance).

You will write a function `ChangeTransaction` that reads in the data from this file into a data frame and changes the amount of a given transaction by adding a specific value. Notice that if you change the amount of one transaction, you also need to change the balance at that transaction, and the balance for every transaction **after it**.

For example, calling the function `ChangeTransaction(2, 10)` returns a data frame with the data above changed to:

```
transactionID amount balance
            1    -10      90
            2    -20      70
            3     10      80
            4    -20      60
```

Notice how transaction 2 had 10 dollars added to the `amount`, and transactions 2 through 4 had 10 dollars added to their `balance`.

Part A (4 marks) On the **next page**, complete the function body of the function `ChangeTransaction` according to its docstring.
You must first read in the data from `account.txt` into a data frame before you change any values (this has been started for you in the function). If you forget how to do this, the back of the quiz has R's built-in functions listed.
Do not use any functions we haven't talked about in class.

Hint: Since the transactionIDs start at 1 and increase by 1, you can make a simple `for` loop sequence based on the given transaction ID.

Continued on next page...

```
ChangeTransaction <- function(transactionID, change) {
  # Returns a data frame that represents the bank account data in a
  # file named "account.txt", in which the transaction at 'transactionID' has
  # its amount changed by adding 'change' dollars to it.
  # All balances are changed accordingly.

  # Read the data from the file into a data frame and assign it to 'accountData'.
  accountData <-

  # Change accountData as needed.









  # Return the changed data frame.
  return(accountData)

}
```

Part B (1 mark)
Given a data frame called `accountData` that was read in from `account.txt`, write an R statement that plots a *line* graph (all other plot settings default) where the x-axis is the **transaction ID** and the y-axis is the **balance** at that transaction ID.
Remember that you are creating a new plot, not adding to a previous one.

Write your code in the box below.

```
>
```

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]

Last Name: _____ **First Name:** _____

R built-in function descriptions

```
paste(s1, s2, sep = s3)
   Returns a new string that is 's1' followed by 's2', with 's3' as the separation character.
   's3' is a space character: " " by default if sep is not specified.
as.double(x)
   Returns a numeric representation of 'x' that is of type 'double'
as.integer(x)
   Returns a numeric representation of 'x' that is of type 'integer'
as.character(x)
   Returns a string representation of 'x' that is of type 'character'.
is.double(x)
   Returns TRUE if and only if 'x' is of type 'double'.
is.integer(x)
   Returns TRUE if and only if 'x' is of type 'integer'.
is.character(x)
   Returns TRUE if and only if 'x' is of type 'character'.
is.na(x)
    Returns TRUE if and only if 'x' is NA.
typeof(x)
   Returns a string that identifies the type of the value of x.
sum(x), prod(x)
    Returns the sum of all elements of 'x', returns the product of all elements of 'x'.
nrow(x)
   Returns the number of rows in 'x'.
ncol(x)
   Returns the number of columns in 'x'.
dim(x)
   Returns a vector containing the number of rows followed by the number of
   columns in 'x'.
append(x, values, after = length(x))
   Returns a copy of set 'x' (vector, list, etc.), which has added to it 'values' directly
   after index 'after'.  'after' is the length of x by default.
rep(x, times)
   Returns a vector of 'x' repeated 'times' number of times.
seq(from, to, by)
   Returns a sequence vector starting at 'from', and ending at or
   before (if it goes over) 'to', incrementing by 'by'.
read.table(filename, header = FALSE)
    Reads the file 'filename' in table format and returns a data frame from it.
    A table formatted file has each column separated by a space.
    If 'header' is set to TRUE, uses first row as
    the header (names of the columns).  'header' is FALSE by default.
plot(x, y, type = "p")
    Plots a new graph where 'x' represents the values on the x-axis, and 'y' represents
    the values on the y-axis.  'type' represents the type of graph, where the
    default value is "p" - a point graph.
runif(x)
   Returns 'x' randomly-generated numbers (uniformly distributed) between 0 and 1 in a vector.
set.seed(x)
    Specifies the seed value used to generate random numbers.
```