SESSION 3: FOUNDATIONAL R PROGRAMMING

Assignment 1

Problem Statement

1. Define an m x n matrix of zeros and then enters a nested-for loop to fill the locations of the matrix, only if the two indexes differ.

```
# Create a 3 x 3 matrix (of 3 rows and 3   columns)
mymat <- matrix(nrow=3, ncol=3)
# For each row and for each column, assign values based on position: product of two indexes
for(i in 1:dim(mymat)[1]) {
  for(j in 1:dim(mymat)[2]) {
    mymat[i,j] = i*j
  }
}
# Just show the upper left 10x10 chunk
mymat[1:3, 1:3]
```

• The purpose is to create a lower triangular matrix, that is a matrix whose elements below the main diagonal are non-zero, the others are left untouched to their initialized zero value.

```
# Lower triangular matrix
make a lower triangular matrix (zeroes in upper right corner)
m=10; n=10;
ctr=0;   # used to count the assignemnt
mymat = matrix(0,m,n) # create a 10 x 10 matrix with zeroes
for(i in 1:m) {
  for(j in 1:n)
  {
    if(i==j)
    {
      break;
    } else
      {
      mymat[i,j] = i*j   # we assign the values only when i<>j
      ctr=ctr+1
      }
  }
  print(i*j)
}
```

• When the indexes are equal (if condition in the inner loop, which runs over j, the column index), a break is executed and the innermost loop is interrupted with a direct jump to the instruction following the inner loop, which is a print; then control gets to the outer for condition (over the rows, index i), which is evaluated again.

```
ttt <- matrix(c("O", "NA", "X", "NA", "O", "O", "X", "NA", "X"),
              nrow = 3,
              byrow = TRUE)

# Define the double for loop
for (i in 1:nrow(ttt)) {
  for (j in 1:ncol(ttt)) {
    print(paste("On row", i, "and column", j, "the board contains",
ttt[i,j]))
  }
}
```

• If the indexes differ, the assignment is performed and the counter is incremented by 1.

```
i <- 1
while (i < 100) {
  print(i)
  i <- i + 1
}
```

• At the end, the program prints the counter ctr, which contains the #number of elements that were assigned.