Assignment 3.1

FOUNDATIONAL R PROGRAMMING-I

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.

• 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.

• 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.
• If the indexes differ, the assignment is performed and the counter is incremented by 1.
• At the end, the program prints the counter ctr, which contains the #number of elements that were assigned.

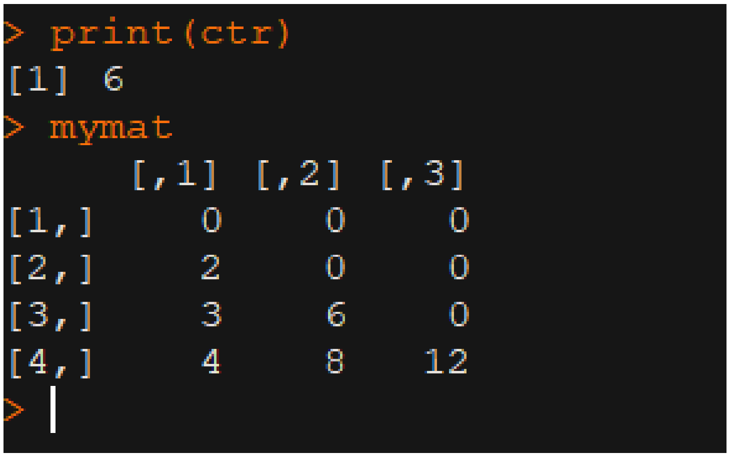
Answer:

- Step 1: Creating an m x n matrix and declaring inputs for **m** and **n.**
- **Step 2:** Storing the matrix in a variable mymat<- matrix(0, m, n)
- Since the matrix is two-dimensional , hence using two for loops for each index and declaring a counter for the count of non-zero value as 0

```
ctr=0
for(i in 1:m){
 for(j in 1:n){
  if(i==j){
   break;
  } else {
   mymat[i,j]= i*j
   ctr=ctr+1
  }
 }
}
print(ctr)
```

- For input **m=4** , **n=3**, the counter (ctr) value is 6. This means 6 non-zero values were added to the matrix to form a **“lower triangular matrix”**

Output: R script is added for the entire code.



```
> print(ctr)
[1] 6
> mymat
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    2    0    0
[3,]    3    6    0
[4,]    4    8   12
> 
```