

## CSC 121 — Lab Exercise 7

This is a non-credit exercise, which you do not hand in.  
You may work on your own or together with another student, as you please.

For this lab exercise, you can try to write the three functions described below. It might also be a good time to work on any previous lab exercises that you didn't finish at the time.

### Find product of elements above the diagonal of a matrix.

Write a function called `multiply_above_diagonal` that takes as its only argument a square matrix (you don't have to check this), and returns a single number that is the product of all the elements in the matrix that are above the diagonal (ie, all those in positions where the column index is greater than the row index). Note that the product of zero factors is defined to be one.

Here is the correct output for two test cases:

```
> Y <- matrix (1:16, nrow=4, ncol=4)
> Y
      [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> multiply_above_diagonal(Y)
[1] 1228500
> multiply_above_diagonal(matrix(5,nrow=1,ncol=1))
[1] 1
```

### Find the length of the longest run in a vector.

Write a function `find_longest_run` that takes one argument that is a vector, and returns the length of the longest “run” in this vector. A “run” is a sequence of consecutive values that are all the same. You can assume that none of the elements of the vector are NA.

For example:

```
> longest_run(c(5,1,1,3,2,2,2,7))
[1] 3
> longest_run(c(5,1,3,3,2,7))
[1] 2
> longest_run(c(8,3,1,3,8))
[1] 1
> longest_run(1:1000)
[1] 1
> longest_run(7)
[1] 1
```

### Swap years born and died entered incorrectly.

Suppose we have a data frame containing data on people, that has columns `born` and `died` that are supposed to be the years that each person was born and died. However, it seems that for some people, these were mistakenly swapped. Write a function `swap_born_died` that swaps these dates for all people for which the year they are said to have been born is later than the year they are said to have died.

For example:

```
> d1 <- data.frame (list(sex=c("M","F","F","M","M"),
+                         born=c(1897,1941,1902,1910,1923),
+                         died=c(1977,1902,1988,1931,1888))
+ )
> d1
  sex born died
1   M 1897 1977
2   F 1941 1902
3   F 1902 1988
4   M 1910 1931
5   M 1923 1888
> swap_born_died(d1)
  sex born died
1   M 1897 1977
2   F 1902 1941
3   F 1902 1988
4   M 1910 1931
5   M 1888 1923
```

You can try solving this using a loop, and without a loop (using vector operations).

Once you have this version working, you can try modifying your function so that it doesn't try to swap `born` and `died` if either of these is NA. You'll need to come up with a data frame with some NA values to test it on.