M.Sc. in High-Performance Computing
5633 - Numerical Methods
Assignment 0

Marina Krstic Marinkovic
(marina.marinkovic@cern.ch)
School of Mathematics, TCD

Rules

This is a non-assessed assignment. If you want your programs checked, make a single tar-ball with all your scripts and a pdf of any written part (or plots) you want to include. Send this via email to `marina.marinkovic@cern.ch` by **Thursday October 6th**, until the uploading system is in place. There will be no marks given for this assignment - however, you will get comments on your implementation. Please note that for late submissions without prior arrangement, feedback will not be provided.

Question

1. Write a function `turned()` that rotates a square matrix, turning it by 90° in anti-clockwise direction:
   (a) using any temporary space you need for the first implementation.
   (b) Then try to do the same minimasing any extra space that you need for implementing the rotation.

   Example of an input matrix from a command line:

   ```
   y<-matrix(1:16, nrow=4,ncol=4)
   ```

   Input matrix y:

   ```
   1 5 9 13
   2 6 10 14
   3 7 11 15
   4 8 12 16
   ```

Output matrix `turned(y)`:

```
13 14 15 16
9 10 11 12
5 6 7 8
1 2 3 4
```

Hint: Imagine you want to write a piece of code that exchanges the values of the two variables `x` and `y`, e.g.

```
x=3
y=5
```

and after the execution of the program, one wants to have:

```
x=5
y=3
```

The code that performs the swap would need an extra variable `t` of the same type as `x` and `y`:

```
t=x
x=y
y=t
```

Therefore, any two numeric variables can be exchanged using a single temporary variable `t`.

Imagine now that x and y are vectors:

```
x=(1,2)
y=(3,4)
```

If you wanted to swap their values, you could either do it by defining a temporary vector tz:

```
tz=x
x=y
y=tz
```

Or by just using an integer variable `t` and swapping one component at the time:

```
t=x[1]
x[1]=y[1]
y[1]=t
```

and then doing the same for the 2nd component of the vector.

Similarly, in order to swap or rotatate a matrix, one could initially use a "temporary matrix", which is not optimal in terms of the memory space used, or try to do the rotation with the minimal space that is needed for the storage of the temporary variables (this type of solution is requested in task `1(b)`).

2. Write a `R` script that reads a table `n`×`m` (`m`≥`3`) from an arbitrary input file `mytable.txt` delimited with spaces `" "` and prints in an output file `myresults.txt` the average value (mean) and standard deviation of each column in the provided table.

   Then extend your script so that it plots a graph where the points on the `x-axis` are the values in the first column of the original table read from the input, the second column of the original table gives `y-axis` values and the third column gives the error bars on the `y-axis`.

   Example of a content of the input file `mytable.txt` :

   ```
   1.2 3.4 5.6 7.8
   2.3 4.5 6.7 8.9
   3.4 5.6 7.8 9.8
   ```

   Please note that the size of an input table in not fixed and your script should work for any table size (`n`×`m`, `m`≥`3`)

   Hint: study pre-defined `R` function `length()`.