CSC 120 (R Section)— Lab Exercise 6

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.

In this lab, you will get more practice in creating and manipulating matrices and lists. You will also learn about the `Inf` value, representing "infinity".

Part I: In this part, you will write a `plot_dots` function that plots points at locations specified by a list of numeric vectors, plus two functions, called `triangle` and `checkerboard`, that produce such lists of vectors that can be passed to `plot_dots`.

The `plot_dots` function takes as arguments a list giving point locations, the horizontal extent of the region where dots are plotted, and the vertical extent of the region where dots are plotted. Its definition should begin with

```
    plot_dots <- function (dots, horizontal, vertical)
```

It should create an emply plot with `xlim` set to `c(0,horizontal+1)` and `ylim` set to `c(0,vertical+1)`, and then add points specified by the dots argument. The dots argument should be a list of numeric vectors. The first numeric vector in this list (which can be gotten by `dots[[1]]`) will contain the y coordinates of points that should be plotted at x coordinate 1, the second vector in dots will contain the y coordinates of points to be plotted with x coordinate 2, etc. Note: The `rep` function might be useful here.

The triangle function should take a single integer, n, as its argument, and return a list with n elements, the first being the vector 1:1, the second 1:2, and so forth up to the last element being 1:n.

The checkerboard function should take a single integer, n, and return a list of numeric vectors that when passed to `plot_dots` will plot a checkerboard pattern of dots on an n by n square (ie, one that has a dot at (i,j) if i+j is an even number).

These functions should be put in a file called dots.R, which can be read with the source function (or by clicking the Source button in RStudio) to make them defined. You can then try them out in the R console, or make another script file that tests them, perhaps like this:

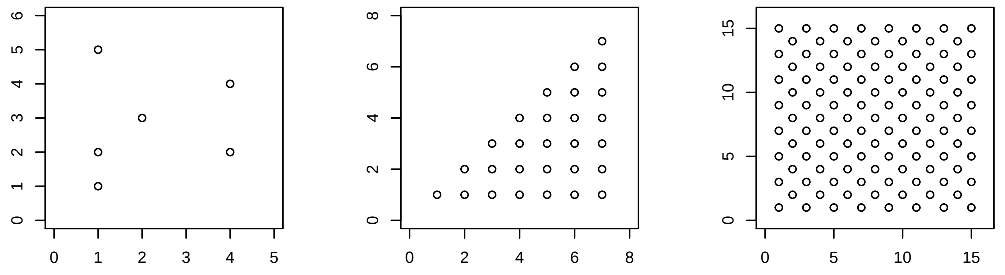
```
    source("dots.R")
    par(mfrow=c(1,3))

    # Test of plot_dots with a small list created manually.
    plot_dots (list (c(2,5,1), c(3), c(), c(4,2)), 4, 5)

    # Test of plot_dots with the triangle function.
    plot_dots (triangle(7), 7, 7)

    # Test of plot_dots with the checkerboard function.
    plot_dots (checkerboard(15), 15, 15)
```

This produces plots something like the following:



Part II:

In this part, you will write a function called `record_lows_and_highs` that takes as its only argument a vector of daily noontime temperatures, and returns a list with two elements, named `lows` and `highs`, that are vectors of temperatures that are record lows or record highs, up to the day when they occurred.

Here is an example call:

```
> record_lows_and_highs(c(5,9,2,-8,-7,10,-12,8))
$lows
[1]   5   2  -8 -12

$highs
[1]  5  9 10
```

Note that the first temperature will always be both a record low and a record high, since there were no earlier temperatures.

There are several ways that you could write this function. One way would be to go through the temperatures in sequence, while using and updating two variables, `record_low` and `record_high`, which record the lowest temperature so far and the highest temperature so far. When the next temperature is below `record_low`, it needs to be added to the vector of record lows, which you can do with the `c` function, after starting with a zero-length vector obtained with `numeric(0)`. You need to start by setting `record_low` and `record_high` to suitable values. For this, it is convenient to use R's `Inf` value, which means "infinity", since any actual temperature will be less than `Inf`. You can also use `-Inf`, which is negative infinity.