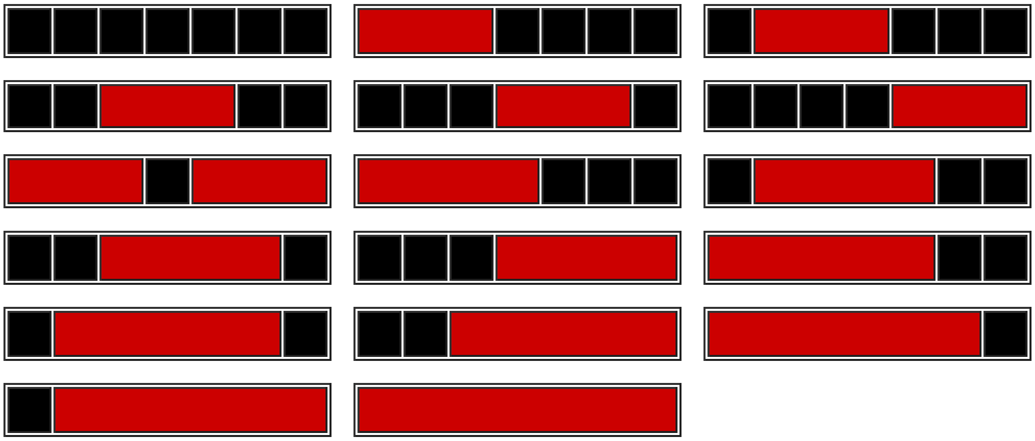
CS 3500 – Programming Languages & Translators
Homework Assignment #6

- This assignment is **due by 8 a.m. on Tuesday, November 14th**
- This assignment will be worth **5%** of your course grade.
- You are to work on this assignment **by yourself**.

Basic Instructions

Suppose that we want a row measuring 7 units in length to have red blocks placed on it such that any 2 red blocks (which are allowed to be of different lengths) are separated by exactly 1 black square. An additional requirement is that a unit of red blocks has to be of size 3 or larger. There are exactly 17 ways of doing this, as shown below.



You are to write a program in **LISP** to **determine the ways a row of a particular length can be filled with red blocks under these conditions**. Specifically, your LISP program must have a function named ***placeRedBlocks*** that takes **one parameter** representing the (integer) size of a row. For example, from within *gcl* on the cslinux machines, we should be able to load a single file containing your LISP code, then type the expression *(placeRedBlocks 7)*, and have it return a **list** of all possible configurations of legal red block placements. Note that your code doesn't have to output (or return) the length of the resulting list of configurations; all we should need to do to find that out, for example, is to evaluate the expression *(length (placeRedBlocks 7))*.

```
(placeRedBlocks 7)
((#\B #\B #\B #\B #\B #\B #\B)
(#\R #\R #\R #\B #\B #\B #\B)
(#\R #\R #\R #\B #\R #\R #\R)
(#\B #\R #\R #\R #\B #\B #\B)
(#\B #\B #\R #\R #\R #\B #\B)
(#\B #\B #\B #\R #\R #\R #\B)
(#\B #\B #\B #\B #\R #\R #\R)
(#\R #\R #\R #\R #\B #\B #\B)
(#\B #\R #\R #\R #\R #\B #\B)
(#\B #\B #\R #\R #\R #\R #\B)
(#\B #\B #\B #\R #\R #\R #\R)
(#\R #\R #\R #\R #\R #\B #\B)
(#\B #\R #\R #\R #\R #\R #\B)
(#\B #\B #\R #\R #\R #\R #\R)
(#\R #\R #\R #\R #\R #\R #\B)
(#\B #\R #\R #\R #\R #\R #\R)
(#\R #\R #\R #\R #\R #\R #\R))
```

```
(length (placeRedBlocks 7))
17
```

We will test your program on a variety of row lengths greater than zero. Some sample results are shown below:

(length (placeRedBlocks 7)) should return 17
(length (placeRedBlocks 8)) should return 26
(length (placeRedBlocks 9)) should return 39
(length (placeRedBlocks 10)) should return 57
(length (placeRedBlocks 11)) should return 82
(length (placeRedBlocks 12)) should return 117

There is a C++ solution for this problem posted on Canvas along with this assignment. You can compile and execute it to check your output for various inputs.

What to Submit for Grading:

Via Canvas you should submit only your *lisp* source code file. Name your file using **your last name followed by your first initial** with file extension **.lisp** (e.g., Homer Simpson would name his file **simpsonh.lisp**). Make sure that your file has a function in it named ***placeRedBlocks*** that expects one integer parameter as explained above; this is what our grading script is going to call! You can submit multiple times before the deadline; only your last submission will be graded.

WARNING: If you fail to follow all of the instructions for submitting this assignment, the automated grading script will reject your submission, in which case it will NOT be graded!!!

The grading rubric to be used for this assignment is given below. **Note that any assignments that "hard code" outputs for specific inputs will receive a grade of 0 for the assignment ☹**

Functionality	Points Possible	Mostly or completely incorrect (0% of points possible)	Needs improvement (50% of points possible)	Mostly or completely correct (100% of points possible)
Implementation - logic	30			
Implementation - syntax	30			
Results - correctness	40			