Report Assignment V

Problem 1:

- Assumptions: The program expects the 2D-array of the board to be user-input in both Mips and Riscv and to be in memory in the x86 program.
- Design Approach:
    a. In this problem implementation of bitwise operators was heavily used both to discover where it is safe to put a lizard in the current row and to mask and umask the effects of a tree
    b. Three primary data structures were used namely ld, rd and col these were initiated and edited as follows:
        1. When storing a one we get the binary representation of the column where the one was stored and put it in all the three data structures
        2. When we recurse to the next row both ld and rd are updated by shifting ld to right by one bit and shift rd left by one bit
        3. The three data structures are anded with a number in binary the size of the board so that any diagonal or anti diagonal effect outside the board is not taken to consideration
    c. There are mainly two functions that recurse the whole map putting lizards this function is called lizards inside it there is a certain for_this_cell function that recurses the whole map starting a given address.
    d. If for_this_cell function couldn't get an address while recursing the whole map then upon return to the lizard function it is given a next address to start from so that the algorithm starts the next time from the second cell in the first row and try to find a solution, before starting again all ones are deleted by the clean_board function.
- Limitations and benefits of the algorithm:
    a. The algorithm is relatively fast due to the implementation of bitwise operators to handle the heavy tasks
    b. The algorithm is set so that whenever it encounters a tree it checks below it and if it is safe it stores one then get back to the row above it, this is a little problematic since the program identifies if the row has a lizard or not by checking a certain marker and since this marker is checked-off when we come from below the tree (given the tree masks effects of lizards in its own row), when we recurse to the new row in which we already put a one we have no way of telling thus a buffer is put in place so that whenever we store one under the tree, this buffer is checked on and its value is transferred to the marker when we recurse to the next row,

thus the program identifies the one; however if there are multiple trees in one row this will still create a problem

c. The program doesn’t offer solutions for all maps there is since for simplicity and performance it only recurse putting lizards in the first row and then adapting the rows below accordingly so whenever all cells in the first row are recursed and no solution was found a message printing no solution is found is printed but again it is something the algorithm lacks.