Prolog and Constraint Logic Programming (clpfd) (35 marks)

In this question, you can use any builtin/pre-defined predicates available in swi prolog unless specified otherwise. Your program should not run into an infinite loop.

1. [6 marks] Define a predicate `odd(+L1,-L2)`, where `L1` is a non-empty input list, and `L2` is the list of all elements of `L1` that are at odd positions, i.e., removing all elements of `L1` at even positions. E.g.,

   ```
   ?- odd([a], L).    ?- odd([a,b],L).    ?- odd([a,b,c,d,e], L).
   L = [a]            L= [a]              L = [a,c,e]
   ```

   In each case, no more answers should be generated when the user types “;”.

2. [6 marks] A palindrome is a word that reads the same backward as forward. Define a predicate `palindrome(L)`, where `L` is a list of letters representing a word, and the goal is proved if `L` is a palindrome and false otherwise. E.g., the following goals should be proved.

   ```
   ?- palindrome([m,a,d,a,m]).

   ?- palindrome([a,n,n,a]).
   ```

   Note that the length of a palindrome can be either even or odd.

3. [9 marks] Define a Prolog predicate `gen_matrix(+N,+M,-Matrix)` that generates an N × M matrix of variables and has variable `Matrix` bound to that matrix. An N × M matrix is represented by a list of N sublists, each of which represents a row and is of length M. E.g.,

   ```
   ?- gen_matrix(3,4,Mtr).
   Mtr = [[X11,X12,X13,X14],[X21,X22,X23,X24],[X31,X32,X33,X34]].
   ```

   Above, we just write Xij for a variable – Prolog uses its own variable naming strategy.

4. [14 marks] Write a CLP(FD) program with the top predicate

   ```
       restricted_matrix(+N,+M,-Mtr)
   ```

   such that a call to it returns a binding to variable `Mtr`, which is an N × M matrix of positive numbers that satisfies the following constraints:

   - The allowable values of an entry of the matrix are between 1 and N (including 1 and N).
   - The sum of each row is less than or equal to N+M.
   - The value 1 is special - in each row, either it does not occur or it occurs only once.

  - The collection of front elements (the first elements) of rows is special - these elements must be distinct.

For example,

```
?- restricted_matrix(2,3,I).
I = [[1, 2, 2], [2, 1, 2]] ;
......
```

Hint: Set up the top predicate correctly - apply the procedure `gen_matrix/3` from the previous question to get clpfd variables in place. Then define each of the last three constraints above. (For part marks, focus on one constraint at a time.)

In case your definition of `gen_matrix/3` does not run correctly, you will not be able to test your solution to this problem. But you may still be able to get part marks by handwriting your code.