1 Problems as Languages

For each of the following problems:

i. Formulate the problem as a *regular* language (give an example of the problem instances and how they are encoded, you don't have to write every problem instance).

ii. Describe the regular expression that describes the expression

Note that how you encode the language matters for the regular expression you end up with.

1. Checking whether (or not) a number is divisible by 4). You are given a binary number and need to output if this number is divisible by 4.
2. The sum of two *unary* integers.
3. The game of TicTacToe. You are given a completed tic-tac-toe board and you need to determine who won. (this won't have a clean regular expression. Just define some encoding and describe how you would build the expression, you don't need to write the whole expression out.) Hint: think about how many games of TicTacToe there are.
4. Given a undirected weighted graph, the shortest path between 2 nodes s and t.

2 Recursive Definitions

Give the recursive definition of the following languages. For both of these you should concisely explain why your solution is correct.

1. A language that contains all strings.
2. A language which holds all the strings containing the substring **`000`**.
3. A language L_A that contains all palindrome strings using some arbitrary alphabet Σ.
4. A language L_B that does not contain either three **`0`**'s or three **`1`**'s in a row. E.g., **`001101`** $\in L_B$ but **`10001`** is not in L_B.