Problem 1:
Write a regular expression for all strings of 0’s and 1’s having an even number of 0’s

\((1*01*01*)*1*\)

The idea is that you need to have 0 and 0 in pair. Then, you need to insert 1* in between, before, and after the pair of 0’s. You could repeat the \((1*01*01*)\) as many as you want. Finally, 1* is used if you don’t have any 0 in the string.

Problem 2:
Write a regular expression for all strings of 0’s and 1’s in which all 0’s occur before any 1’s

\(0*1*\)

This is obvious that all 0’s must appear before any 1’s.

Problem 3:
Write a regular expression for all strings of 0’s and 1’s that are of odd length.

\(((0+1)(0+1))*(0+1)\)

The substring \(((0+1)(0+1))\) are always even; therefore, by appending it with one more \((0+1)\), the string will always be odd.