Question 2.

a) Does Not Execute the fault:

    EMPTY Array, it will result in 0 and will not proceed forward with the rest of the code. Therefore,
    NOT executing the fault. only doing for loop check in empty array for x.length.
    X.length helps not executing the if condition and exits at for loop.

b) Executes the FAULT, but NOT result in ERROR STATE.

    When the INPUT DOES NOT CONTAIN ANY NEGATIVE ODD INTEGERS that result in a "-ve"
    value as a result.
    The same test case can be can be used when -99 -> 99 is done as an example.
    Fault is executed but not an error state observed.
    eg: [1,2,67,-16]

c) Results an ERROR, but NOT A FAILURE.

    When the input is a NULL array, it results in an ERROR but not a failure.
    The error is NullPointerException and it does not executes as a failure.
    The code exits the method at that point.

d) First ERROR State, with complete description of the state.

    input: x = [-18, 0, -99, 17, 102, 16]
    Expected output: 4
    Actual output: 5
    First Error State:
        x = [-18, 0, -99, 17, 102, 16]
        i = 2
        count = 0
        PC: i++

    This is where the fault occurs as -99%2 = -1 and is not being counted towards
    an odd number. As done in the Class notes we see that after doing the
    execution of if(x[2]%2 == 1) it does not go inside but jumps to the for loop and
    that next value of program counter is the Error state.