Question 1

1-)

it says in the home work that input n will always be in the type of 3^k it is not possible for n to be even number so it can't enter the else if(n%2 ==0) statement because the remainder is never 0 when divided by 2, so that else if is unnessesary.

let's say the function that calculate the number of stars when it's given a number "n" (which is in 3^k format) is T(n), when we say that we see $T(3^0) = T(1) = 1$ this is the base case $T(1) = 1$

when the unnessoryStars(n) function enters the (n%2 ==1) else if statement we can see it enters the recursion which prints 1 star for each recursion in the amount of times the n is divisible by 3 until it reaches to the 1; it prints 3 stars when each division step has ended, so we can express the T(n) as $T(n) = 3.T(\frac{n}{3}) + 3$ except $n \neq 1$.

2-) to find the closed form expression for the reccurence we will write the successive expansions.

1. $T(n) = 3.T(n/3) + 3$, lets put put the T(n) inside the "n" in $T(n/3)$

2. $T(n) = 3.(3.T(n/9) + 3) + 3$, lets do it again

3. $T(n) = 3^2.(3.T(n/27) + 3) + 3^2 + 3$, now we can see the pattern, the closed form expression would be;

after k amount of steps the closed form expression would be;

$T(n) = 3^k . T(n/3^k) + 3^k + 3^{k-1} + \dots 3^1$ $\sum_{n=1}^{k} 3^k = \frac{1-3^k}{1-3}$

since it is said that $n = 3^k$ we can solve the k and find $k = \log_3 n$. when we input $n = 3^k$ and $k = \log_3 n$. We can get rid of the geometric series because it converges to a constant value and for $3^k . T(n/3^k)$ it also converges so the whole T(n) converges to a constan so the time complexity can be expressed as O(n)

$T(n) = \Theta(n)$, so we expressed the result as O(n) by solving the reccurrence by successive expansions,

Question 2

1-) Since this question is almost the same as Question 1 i will do less explanations.
let's define $T(n)$ which gives the number of zeros that will be printed when given a 2^k type of n.
our base case is $T(0) = 1$.

printZerosHelper prints $(\log_2 n) + 1$ amount of zeros.
and there is two reccurence of $T(n-2)$ so
$T(n) = (\log_2 n + 1) + 2 \cdot T(n-2)$.

2-)
1. $T(n) = (\log_2(n) + 1) + 2 \cdot T(n-2)$
2. $T(n) = (\log_2(n) + 1) + 2 \cdot ((\log_2(n-2) + 1) + 2 \cdot T(n-4))$
3. $T(n) = (\log_2 n + 1) + 2 \cdot (\log_2(n-2) + 1) + 4 \cdot ((\log_2(n-4) + 1) + 2 \cdot T(n-6))$

So if we do this for k steps.
After some amount of calculations we see that the common ratio of this geometric series is 2 which can be simplified as $2^{n/2} \cdot \log_2(n)$ so the $T(n) = O(2^{n/2} \cdot \log_2 n)$ time complexity.

2.1

Index of comments

2.1 You should have mentioned how you got n/2. What was the termination condition.