**Answer to Question 1**  
  
1) Let S(n) be the number of stars printed by the program, and for some k.

The second else if block will never be entered as is never equal to an even number.

The base case (k = 0, n = 1) prints 1 star, so we can say:  
S(n) = { 1 , if n = 1  
 ? else}

Now we can go on to analyze the rest of the cases, namely when k > 0.  
When k = 1, then program is called recursively 3 times within the second else if block, and then a helper function (printStarsHelper() prints 3 stars) is called. Thus, we have stars printed in the case of S(3).

We can then by induction conclude that our generalized formula for S(n) is:

2)

T S(n)  
1

2

3

. .

. .

. .

t   
General Form

End of recurrence is reached when n = 1, hence, to make equal to 1, we substitute t by .

Hence our equation is  , which is of complexity .

**Answer to Question 2**1)   
Let T(n) be the number of zeros printed by the unnecessaryZeros(n) function.

Base case:

T(0) = 1, as we print a single "0" when n is zero.

Recursive cases:

If n > 0 is even, then we call printZerosHelper(n), which prints zeros, and then recursively call unnecessaryZeros(n-2) twice. Therefore:

, for n > 0 and even

n > 0 can only be odd for n = 1, in which case only 1 zero is printed:

Hence our generalized formula would be:

2)

k T(n)  
1

2

3

4

. .

. .

k

Recurrence ends when , which is when , thus .

Substitute k by n/2:

We can omit as it’s a coefficient of :

Since the dominating term in this expression is ,

Running time complexity of our function is .