Assignment 11

1) Rewrite the following function into Lambda syntax:  
function largestOfThree(a, b, c) {  
if (a > b && a > c) return a;  
if (b > a && b > c) return b;  
return c;  
}

const largestOfThree = (a, b, c) ==> {

if (a > b && a > c) {return a};

if (b > a && b > c) {return b};

{return c};

}

2) Rewrite your previous program that called an API so that it uses lambda syntax.

See attached

3) Explain in English how this recursive function runs:  
function sum(n) {  
if (n === 1) {  
return 1;  
}  
return n + sum(n - 1);  
}

A number is input as a variable to function called sum as (n). For example, sum(5). The if statement defines the last byte so the function terminates and returns the final answer in return 1. Otherwise the function reiterates over each number. For example, return 5 (n) + sum(5-1)

5 + 4

9 + 3

12 + 2

14+1

Final answer returned is 15

Sum(5) is called. It then checks is 5 is equal to 1 which is no. Then it returns 5 plus sum(4).

1. When you call sum(5):
   1. It checks if 5 is equal to 1 (no).
   2. Then it returns 5 + sum(4).
2. The call to sum(4):
   1. Checks if 4 is equal to 1 (no).
   2. Returns 4 + sum(3).
3. This process continues until the call reaches sum(1) where it returns 1 directly.
4. The function then calculates the sum by adding each value back up the call stack: 1 + 2 + 3 + 4 + 5 = 15.

4) Do some research on Google etc. and explain in English why calling the sum function with a  
very large input number fails - e.g. sum(11111) fails with error: Maximum call stack size  
exceeded

The function calls itself without any condition to eventually stop and the program runs out of space to keep track of the repeated calls because the stack memory is limited to a certain amount of bytes.