Question 1

Undefined//console.log(x) cause there’s an x inside var c but it’s called before initializing it.

8//console.log(a) cause in the function scope there’s “a” in the parameters.

8//Console.log(b) inside var f the variable b will take the value of a which is 8.

9// console.log(b) cause the changes that occurred to b inside var f don’t concern the variable b that’s outside of it.

10// cause the local value of b is 10

1// cause the assignments that were made to x only concerned the var x that was created inside the function.

Question 2

Local scope is the scope that’s delimited by a block such as a function or a loop.

The global scope is the highest scope. Elements in the global scope can’t access let variables declared in local scopes.

Question 3

1. No, because elements defined inside a function are only accessible inside of it.
2. Yes, elements inside a scope can always have access to statements in scopes that englobe it
3. No statements in a scope can only access statements in the same scope or in scopes that englobe it.
4. Yes for the same reason.
5. Yes for the same reason.

Question 4

81// because myFunction can access variable x;

25//because the value of x changed

Question 5:

1 // cause the the block inside If won’t execute cause !1 is false

+\s[0-9]{3}\s[0-9]{3}[0-9]{4} | [0-9]{10}