Q1

1.Call Stack:

Call stack is the box or thing executes all the functions in the program by line by line and executes the functions in the order they are invoked in a LIFO manner.

2.Event Loop:

Event loop is also a place like structure where settimeout and setInterval like algorithms goto get their time finished and then pushed into callstack by completing their amount of time needed the executed and by this way it plays asynchronous processing.

3,4.setTimeout and Promises fit into the Event Loop:

Consider this code has both new promise and setTimeout:

1. function fun(){

2.return new promise((res,rej)=>{

3.setTimout(()=>{

4.res(“Promise resolved”);

5.},1000);

6.});

7.setTimeout(()=>{

8.console.log(fun());},5000);

In this code while execution ,it goes by by this order as asynchronous:

1.the execution comes to line no7 after creating GEC

2.then setTimeout is pushed into call stack ,it will not be handled there so it will move to WEB API to perform timing then to callback queue by order of FIFO the callbacks are pushed to call stack by EVENT LOOP then executed then to function and then promise the resolved message

Logged in console.

