1. demo2.js:

答案：1 3 2

原因：setTimeout(),属于异步，在执行时会先加入队列，

代码同步执行（从上到下），当执行到setTimeout ()时会先把里面的回调函数放进队列，然后继续往下执行代码，当同步执行完以后，在循环队列，从而执行定时器里面的回调函数，所以结果就是：1 3 2；

1. demo3.js:

答案：2 1

原因：原理和1一样。

1. demo4.js:

答案：1 2 333

原因：process.nextTick（）总是在当前执行站的尾部触发（当同步执行完以后（队列执行前）执行它里面的语句）；所以console.log(1); 要优先于process.nextTick()执行；所以最终结果是 1 2 333