MWA - Homework

You are in a technical interview for a fullstack engineer position:

1. The first question you will have to answer is explaining JS event loop, clarifying the synchronous and asynchronous parts, queues, priority, with examples.

Write your answer as if it were for a real interview.

2. The follow up question is how may we convert a sync operation/function to become asynchronous?

Question 1:

The NodeJs event loop is what allows NodeJs to execute non-blocking I/O operations by the continuous scheduling and dispatching of events from the macro task queue, or micro task queue into the execution call stack whenever the stack is empty.

Every synchronous code is executed directly by the call stack in a sequential manner, while every asynchronous code encountered in the code block (either the global scope or function scope) is then scheduled into the Web API of the JavaScript engine residing in the browser, and based on the first-come, first-serve model, these asynchronous codes are then loaded onto the macro task queue while awaiting to be dispatched by the event loop into the execution call stack.

In place of priority, time is the only differentiating factor that determines which asynchronous code gets queued first or not. But, it is expedient to note that the micro task queue has a higher execution priority than the macro task queue, so the event loop will dispatch events from the macro task queue only if there are no events in the micro task queue and the execution call stack is empty.

Question 2:

You can convert a synchronous operation/function to an asynchronous in several ways, and the most basic will be to put the operation/function into a setTimeout or setInterval event. You can also convert the operation/function into a promise and resolve or reject the return value of the promise executor.