1.How many callbacks queues in Node.js and what are they? Give examples to explain how different callbacks are enqueued in different queues.

Ans : Node.js has 6 callback queues 2 microtask queues called nextTick queue and promise queue and 4 other macrotask queues called timer queue , IO queue , check queue and closer queue

In node.js when we run our code the async non blocking codes are added to the call stack and then removed to be added on their respective queue. The event loop start running once the call stack is empty.

Here’s a general algorithm of how code is executed on the event loop

1. Any callbacks in the micro task queues are executed. First, tasks in the nextTick queue and only then tasks in the promise queue
2. All callbacks within the timer queue are executed
3. Callbacks in the micro task queues if present are executed. Again, first tasks in the nextTick queue and then tasks in the promise queue
4. All callbacks within the I/O queue are executed
5. Callbacks in the micro task queues if present are executed. nextTick queue followed by
6. Promise queue.
7. All callbacks in the check queue are executed
8. Callbacks in the micro task queues if present are executed. Again, first tasks in the nextTick queue and then tasks in the promise queue
9. All callbacks in the close queue are executed

2.
Output will be :

```
start
end
nextTick 1
nextTick 2
Promise...1
Promise...2
timeout 1
timeout 2
nextTick 3
timeout 3
```

3.
When running settimout with an argument of 0 the result is not guaranteed hence at times it can be as follows

```
Immediate
timeout
readFile....
```

Or

```
timeout
Immediate
readFile....
```

4. Running code on a browser would output hello world because declaring var in the global would save the variable in the global window so when we access this in the function it will refer to the window which has the variable message.

Running code on node would return undefined

5. This wouldnt work because getName is not a function in the context of app.js so we cant make a function call to it.

To make it work we'll call getName from getName like this

```
getName.getName();
```

6.

First error

Apps is importing pattern2 it should be pattern1

Second error

Even though exports and module.exports point to the same empty object at fist if we at any point reassign exports to another variable it will lose its reference to that object in the heap memory and only module.export will be reterend

In this case app.js will return Josh Edward

7.

Samething as quesiton 6; app.js will return Josh Edward but this time we dont need to fix the code.

8.
IIFE is a function that is immediately invaded right after its creation. Syntax to create an iife function is to wrap a function with () and not including a variable declaration.
For the code snippet the output will be an object containing two attributes first name and last name