

1. What is LibUV?

- is a multi-platform support library with a focus on asynchronous I/O

2. Explain the difference between `setImmediate(f)` and `setTimeout(f, Time)`?

**setTimeout** schedules a callback to run after a specific time, the functions are registered in the **timers phase** of the event loop.

**setImmediate** schedules a callback to run at **check phase** of the event loop after IO events' callbacks.

3. Explain the difference between `process.nextTick(f)` and `setImmediate(f)`?

- `process.nextTick(f)` is not part of the event loop; it adds the callback in the nextTick queue.

- Node processes all the callbacks in the nextTick queue after the current operation completes and before the event loop continues .

- This means `process.nextTick(f)` runs before any additional I/O events or timers fire in subsequent ticks of the event loop.

- `setImmediate` schedules a callback to run at the check phase of the event loop after IO events' callbacks.

this is `setTimeout`

2. Pls write down the output without executing the following code snippets and check it with result.

```
const fs = require('fs');
```

```
//you may assume input.txt is in the same folder
```

```
const rd = fs.createReadStream("input.txt");
```

```
rd.close();
```

```
rd.on("close", () => console.log('readableStream close event'))
```

```
fs.readFile('input.txt', "utf-8", (error, data) => {
```

```
  if (error) console.log(error);
```

```
  else console.log(data));
```

```
setTimeout(() => console.log("this is setTimeout"), 5000);
```

```
setTimeout(() => console.log("this is setTimeout"), 0);

setImmediate(() => console.log("this is setImmediate 1"));

setImmediate(() => {

  console.log("this is setImmediate 2")

  Promise.resolve().then(() => console.log('Promise.resolve inside setImmediate'));

});

Promise.resolve().then(() => console.log('Promise.resolve 1'));

Promise.resolve().then(() => {

  console.log('Promise.resolve 2')

  process.nextTick(() => console.log('nextTick inside Promise'));

});

process.nextTick(() => console.log('nextTick 1'));
```

## Output

```
nextTick 1
Promise.resolve 1
Promise.resolve 2
nextTick inside Promise
this is setTimeout
this is setImmediate 1
this is setImmediate 2
Promise.resolve inside setImmediate
readableStream close event
data from input.txt
```