Lab 12 - answers

1. Pls answer the following:

1. **What is LibUV?**
   LibUV is a multi-platform C library that provides Node.js with:
   - Asynchronous I/O operations
   - Event loop
   - Thread pool
   - File system access
   - DNS resolution
   - Timers
   - TCP/UDP sockets
   - Child process handling

   - It abstracts away OS-level differences, making Node.js code work consistently across Windows, macOS, and Linux.
   - LibUV is the backbone of Node.js's non-blocking, event-driven architecture.
2. **Explain the difference between `setImmediate(f)` and `setTimeout(f, Time)`?**

Aspect	`setImmediate(f)`	`setTimeout(f, Time)`
Execution time	Executes **after** the current poll phase of the event loop	Executes **after** a minimum of ~1ms delay (even if 0 is given)
Priority	Usually executes **sooner** than `setTimeout(f, 0)`	Slightly delayed due to timer granularity
Use case	When you want to run code immediately after I/O callbacks	When you want a scheduled delay

Summary: setImmediate() runs after the current I/O, while setTimeout() runs after the timer queue delay.

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

Aspect	`process.nextTick(f)`	`setImmediate(f)`
Timing	Executes **before** the event loop continues	Executes **after** the current poll phase
Phase	Runs in the **microtask queue**	Runs in the **check phase** of the event loop
Priority	**Higher** than `setImmediate()` — runs immediately after the current operation	Lower — scheduled in the next event loop iteration
Risk	Overusing can **block** the event loop	Safer for deferring large tasks

Summary:

- `process.nextTick()` → before anything else after current operation
- `setImmediate()` → after I/O but before `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('readablStream 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
readablStream close event
<Hello World! – data from input.txt>
this is setTimeout