

Task 1

Code

```
1  setTimeout(() => {
2    console.log('timer')
3    Promise.resolve().then(() => console.log('promise1'))
4  }, 0)
5
6  Promise.resolve().then(() => {
7    console.log('promise2')
8    Promise.resolve().then(() => console.log('promise3'))
9  })
10
11 console.log('script')
```

Output

```
1  script
2  promise2
3  promise3
4  timer
5  promise1
```

Explanation

- `setTimeout` gets called first and adds the anonymous function to the queue
- `Promise.resolve()` gets called next and adds the anonymous function to the end of the stack
- `console.log` gets called and emits the first output
- Next, the Promise is on top of the stack, and `console.log('promise2')` is called
- `Promise.resolve()` gets called next and adds the anonymous function to the end of the stack
- On top of the stack, now there is `() => console.log('promise3')`
- While the stack is now empty, we still have items in the queue: `console.log('timer')` gets called
- Finally, `Promise.resolve().then(() => console.log('promise1'))` is called