Homework#5#

(Synchronous & Asynchronous)

```
console.log('1');
console.log('2');
console.log('3');
console.log('4');
```

```
console.log('1');
console.log('2');
setTimeout(() => {
console.log('3');
}, 2000);
console.log('4');
```

1.) จงอธิบาย Codeทั้งสอน แบบใดเป็น Synchrononus และ แบบใดเป็น Asynchronous เพราะอะไร จงอธิบาย

รูปที่ 1 เป็นแบบ Synchrononus คือ Process ที่จำเป็นตองรอให้ทำงานเสร็จก่อนถึงจะไปทำ Process ถัดไปได้

รูปที่ 2 เป็นแบบ Asynchronous Process ที่สามารถข้ามไปทำ Process ถัดไปได้ทันทีโดยไม่ต้องรอ

```
const fn1 = (value, cb) => {
  console.log('fn1', value);
  cb(value + 1);
};

const fn2 = (value, cb) => {
  console.log('fn2', value);
  cb({
    a2: value + 1,
    b2: value + 2,
  });
};
```

```
const fn3 = (value1, value2, cb) => {
  console.log('fn3', value1, value2);
  cb(value1 + value2);
};

const fn4 = (value, cb) => {
  console.log('fn4', value);
cb();
};
```

```
const main = () => {
  const a = 0;
  fn1(a, (a1) => {
    fn2(a1, ({a2, b2}) => {
      fn3(a2, b2, (a3) => {
        fn4(a3, () => {
         console.log('End');
        });
      });
    });
  });
};

main();
```

2.) จงแก้ไข Code Callback Hell ต่อไปนี้โดยใช้ Promise และแก้ไข Code Promise ด้วยAwait

Promise

```
const fn1 = (value) => new Promise((resolve) => {
  console.log('fn1', value);
  resolve(value + 1);
});
const fn2 = (value) => new Promise((resolve) => {
  console.log('fn2', value);
  resolve({
    a2: value + 1,
    b2: value + 2
  });
});
```

```
const fn3 = (value1, value2) => new Promise((resolve) => {
  console.log('fn3', value1, value2);
  resolve(value1 + value2);
});
const fn4 = (value) => new Promise((resolve) => {
  console.log('fn4', value);
  resolve();
});
```

```
const main = () => {
  const a = 0;
   fn1(a)
    .then((a1) => fn2(a1))
    .then(({a2, b2}) => fn3(a2, b2))
    .then((a3) => fn4(a3))
    .then(() => console.log('End'));
};

main();
```

Async Await

```
const fn1 = (value) => new Promise((resolve) => {
    console.log('fn1', value);
    resolve(value + 1);
  });

  const fn2 = (value) => new Promise((resolve) => {
    console.log('fn2', value);
    resolve({
      a2: value + 1,
      b2: value + 2
    });
  });
```

```
 const fn3 = (value1, value2) => new Promise((resolve) => {
   console.log('fn3', value1, value2);
   resolve(value1 + value2);
 });

 const fn4 = (value) => new Promise((resolve) => {
   console.log('fn4', value);
   resolve();
 });
```

```
const main = async () => {
  const a = 0;
  const a1 = await fn1(a);
  const { a2, b2 } = await fn2(a1);
  const a3 = await fn3(a2, b2);
  const a4 = await fn4(a3);
  console.log('End');
};

main();
```