```
//// rest/sprate operator

var fn = function (...x) {

    //   console.log(x); // result [1,2,3]

    //   console.log(...x);
    // result :
    //1
    //2
    //3

    x.forEach(data => {
        console.log(data + 1)
    })

    // result :
    //1
    //2
    //3

}

fn(1, 2, 3)


////////////////////////////////////// Promise

var p1 = new Promise((resolve, reject) => {
    setTimeout(function () {
        console.log('hello 1')
        reject(100)
    }, 5000)
})

//   var s = await p1;

var p2 = new Promise((resolve, reject) => {
    setTimeout(function () {
        console.log('hello 2')
        resolve(200)
    }, 7000)
})

// p1.then(function(result) {
```

```
//         console.log(result-90)
//         }).catch(err => {
//   console.log(err)
// })

// Promise.all([p1,p2]).then(data => {
//   console.log(data)
// })

Promise.race([p1, p2]).then(data => {
    console.log('Inside then block ' + data)
}).catch(err => {
    console.log('Inside Catch block ' + err)
})



////////////////////// Async/Await

var fn = async () => {
    await setTimeout(function () {
        console.log('hello 1')
        return 1000
    }, 5000)
}

// var fn = function() {
//   setTimeout(function() {
//     console.log('hello 1')
//     return 1000;
//   },5000)
// }

console.log(await fn());


////////////////////// IIFE
//es6
(() => {
    console.log('hello 1')
})();


//es 5
((function () {
```

```
    console.log('hello 2')
})())
```