我们知道 promise.all 要保证了三个 promise 全部执行完毕后，一起输出三个 promise 的结果。

但 11-19 中 采用 if (index===promises.length - 1) 来判断三个 promise 是否 都执行完成，实际是存在一个小问题，问题是：如果最后一个 promise 最先达到了，那必定会出现 先返回，那整个 all 方法就会执行结束。解决这个问题其实很简单，重新定义变量来计算索引就解决问题了，参见下方代码：

```
static all(promises: Promise[]): Promise {

    return new Promise((resolve, reject) => {
      let executorIndex = 0// 重新定义变量来计算索引
      let allPrmiseResolveSucssValue: Array<any> = []
      promises.forEach((promise, index) => {
        promise.then((resolveSuccess) => {
          ProcessData(resolveSuccess, index)
        }, (rejectFail) => {
          reject(rejectFail)//只要有一个Promise2对象的resolve执行失败，执行reject
          return;})})
      function ProcessData(resolveSuccess: any, index: number) {
        allPrmiseResolveSucssValue[index] = resolveSuccess
        executorIndex++
        // if (index===promises.length - 1){
        if (executorIndex === promises.length) {//所有的Promise2对象resolve函数全部执行完成
          resolve(allPrmiseResolveSucssValue)
        }
      }
    })
  }
```