//The Promise object represents the

// eventual completion (or failure)

// of an asynchronous operation )

function tatkalBook(){

return new Promise((resolve,reject) => {

let bookingSuccess = false

if (bookingSuccess)

resolve(850)

else

reject()

})

}

// tatkalBook().then((amt)=>console.log(`Thanks buddy! I have trasferred Rs.${amt}`))

// .catch(()=>console.log("Thanks for trying! I will book a bus"))

function tossCoin(){

return new Promise((resolve,reject)=>{

//0-head(success) 1 -tail(failure)

const rand = Math.floor(Math.random()\*2)

if(rand==0)

resolve()

else

reject()

})

}

// tossCoin()

// .then(()=>console.log("Congrats!Its head!You won"))

// .catch(()=>console.log("Sorry!You lost!Its"))

let reachA = new Promise((resolve,reject)=>{

const reached = true

if(reached)

setTimeout(resolve,3000,"Vidya reached")

else

reject("Vidya not reached")

})

let reachB = new Promise((resolve,reject)=>{

const reached = true

if(reached)

setTimeout(resolve,1000,"Ramya reached")

else

reject("Ramya not reached")

})

let reachC = new Promise((resolve,reject)=>{

const reached = true

if(reached)

setTimeout(resolve,2000,"Latha reached")

else

reject("Latha not reached")

})

Promise.all([reachA,reachB,reachC])

.then((message)=>console.log(message))

.catch((message)=>console.log(message))

//promise - pending,resolved,rejected (settled)

Promise.allSettled([reachA,reachB,reachC])

.then((message)=>console.log(message))

.catch((message)=>console.log(message))

Promise.any([reachA,reachB,reachC])

.then((message)=>console.log(message))

.catch((message)=>console.log(message))

Promise.race([reachA,reachB,reachC])

.then((message)=>console.log(message))

.catch((message)=>console.log(message))