1) How do you define an asynchronous function in JavaScript?
Provide examples of defining asynchronous functions using `async` and `await` keywords.

→ Asynchronous functions are used when you are working with promises. To define an asynchronous function you have to use the ‘async’ keyword while defining a function. For example :

```
async function FetchData ()
{
        const response = await fetch(‘url’);
        return response;
}
```

2) What are Promises and how are they used in JavaScript?
Explain the concept of Promises and their role in managing asynchronous operations.

→ Promises in javascript help in handling asynchronous code, by avoiding the need for nested callback functions.

A Promise can be in one of three states:

1. **Pending**: The initial state; the operation has not yet completed.
2. **Fulfilled**: The operation completed successfully, and the promise is resolved with a result.
3. **Rejected**: The operation failed, and the promise is rejected with an error.

```
const promise = new Promise((resolve,reject)=>{

  setTimeout(()=>resolve("Done"),2000);

})



async function doWork (){

  console.log("doing");

  const res=await promise

  console.log(res)

}
```

```
doWork()
```

How would you handle asynchronous loading of game assets (e.g., images, sounds) in JavaScript?
Provide code examples for loading game assets asynchronously using Promises or async/await.

```
function loadImage(src){
    return new Promise((resolve, reject) => {
    const img = new Image();
    img.onload=()=>resolve(img);
    img.onerror=()=>reject("Failed to load image from path "+src);
    img.src=src;
    })
}

export async function LoadImageAsync(src,Object){
    try{
      Object.image = await loadImage(src);
    }
    catch{
      console.log("Error while loading the image")
    }
  }
```