```
// Assignment 1: sumAsync
function sumAsync(num1, num2, callback) {
   setTimeout(() => {
      const sum = num1 + num2;
      callback(sum);
   }, 1000);
}

// Usage
sumAsync(5, 10, (result) => {
   console.log("Sum:", result); // Output after 1 second: Sum: 15
});

// Assignment 2: getData
function getData() {
   return new Promise((resolve) => {
      setTimeout(() => {
         resolve("Data fetched successfully.");
      }, 2000);
   });
}

// Usage
getData().then((message) => {
   console.log(message); // Output after 2 seconds: Data fetched successfully.
```

```
});

// Assignment 3: fetchData
async function fetchData(url) {
    const response = await fetch(url);
    const data = await response.json();
    return data;
}

// Usage (replace 'YOUR_API_URL' with an actual URL)
// fetchData('YOUR_API_URL').then(data => console.log(data));

// Assignment 4: fetchData (similar to Assignment 3)
async function fetchData(url) {
    const response = await fetch(url);
    const data = await response.json();
    return data;
}

// Usage (replace 'YOUR_API_URL' with an actual URL)
// fetchData('YOUR_API_URL').then(data => console.log(data));

// Assignment 5: multiplyWithCallback
function multiplyWithCallback(arr, callback) {
    const result = arr.map(num => num * 2);
```

```
    callback(result);
}

// Usage
multiplyWithCallback([1, 2, 3, 4], (result) => {
    console.log("Multiplied Array:", result); // Output: Multiplied Array: [2, 4, 6, 8]
});

// Assignment 6: fetchUserDataAndPosts
async function fetchUserDataAndPosts(userId) {
    const userResponse = await fetch(`https://jsonplaceholder.typicode.com/users/${userId}`);
    const userData = await userResponse.json();

    const postsResponse = await fetch(`https://jsonplaceholder.typicode.com/posts?userId=${userId}`);
    const postsData = await postsResponse.json();

    return {
        user: userData,
        posts: postsData,
    };
}

// Usage (replace 'USER_ID' with an actual user ID)
// fetchUserDataAndPosts(USER_ID).then(data => console.log(data));
```

```
// Assignment 7: fetchMultipleData
async function fetchMultipleData(urls) {
   const responses = await Promise.all(urls.map(url => fetch(url).then(res => res.json())));
   return responses;
}

// Usage (replace with actual URLs)
// fetchMultipleData(['url1', 'url2']).then(data => console.log(data));

// Assignment 8: racePromises
function racePromises(promises) {
   return Promise.race(promises);
}

// Usage
const promise1 = new Promise((resolve) => setTimeout(() => resolve("First Promise"), 1000));
const promise2 = new Promise((resolve) => setTimeout(() => resolve("Second Promise"), 500));

racePromises([promise1, promise2]).then(result => {
   console.log("Race Result:", result); // Output: Race Result: Second Promise
});
```