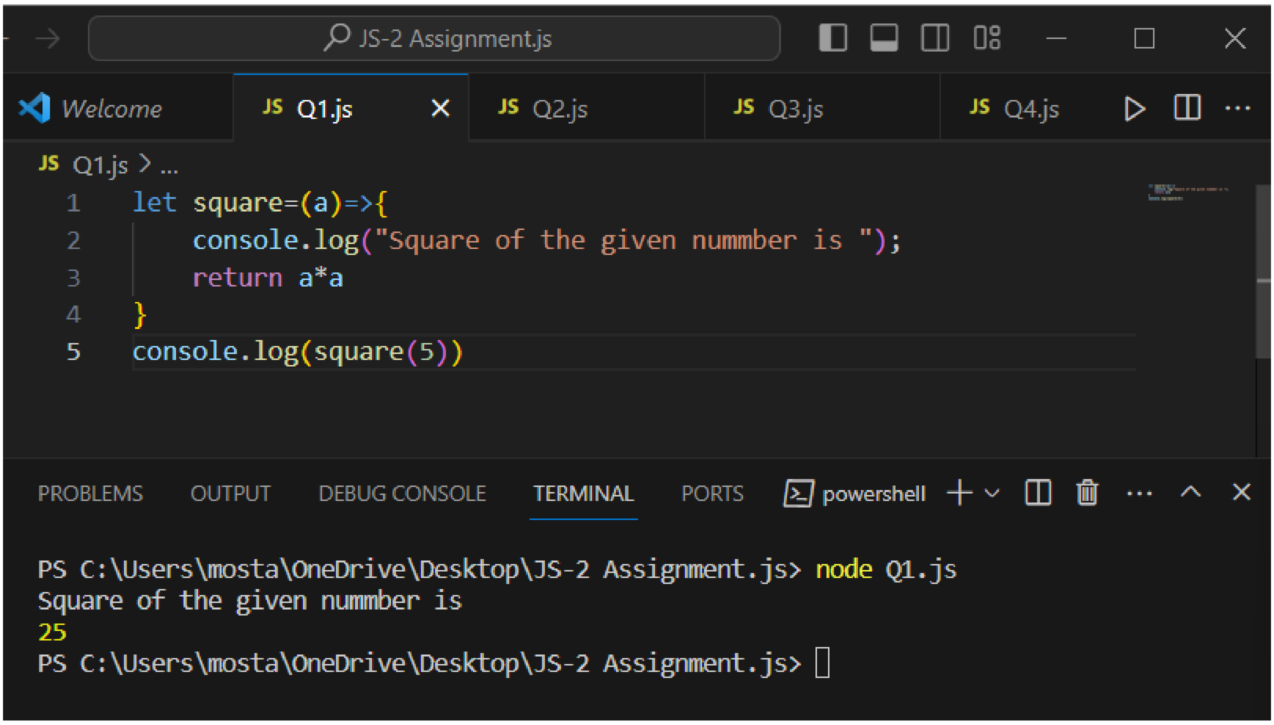
Core Javascript-2

Assignment Solutions

1. Create an arrow function called square that takes a number as an argument and returns its square. Use the arrow function to calculate the square of a given number and display the result.

Solution:-



```
let square=(a)=>{
    console.log("Square of the given nummber is ");
    return a*a
}
console.log(square(5))
```

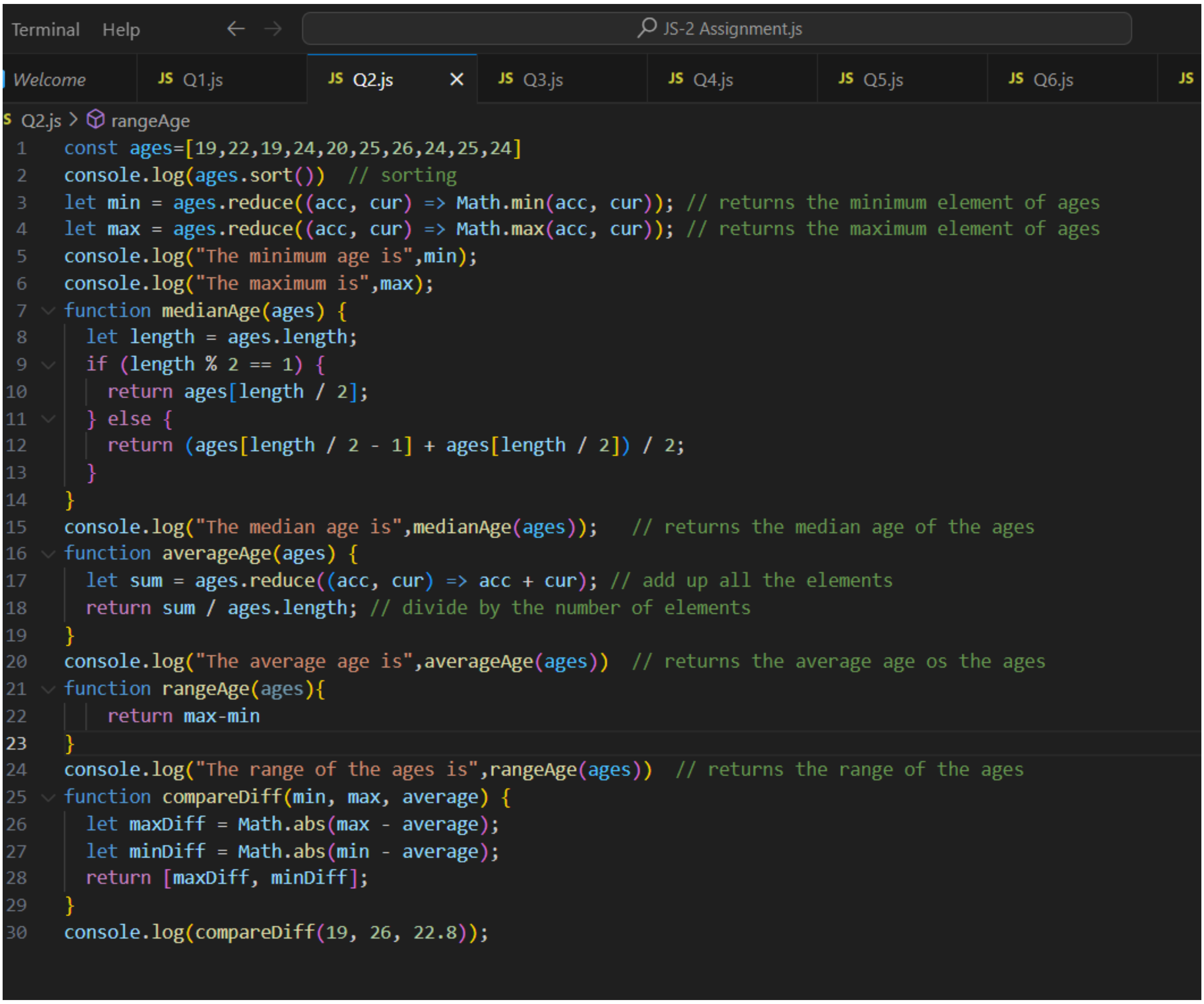
```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q1.js
Square of the given nummber is
25
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js>
```

2. The following is an array of 10 students ages:

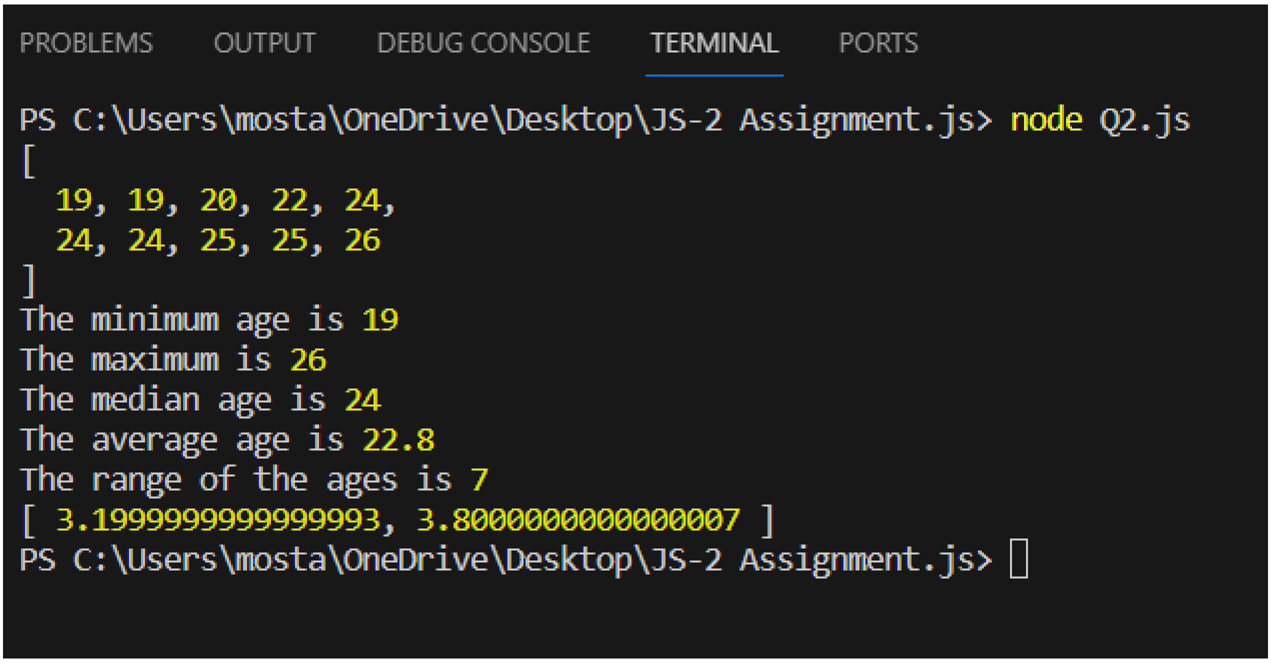
=> const ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

- **Sort the array and find the min and max age.**
- **Find the median age(one middle item or two middle items divided by two)**
- **Find the average age(all items divided by number of items)**
- **Find the range of the ages(max minus min)**
- **Compare the value of (min - average) and (max - average), use abs() method**

Solution:-



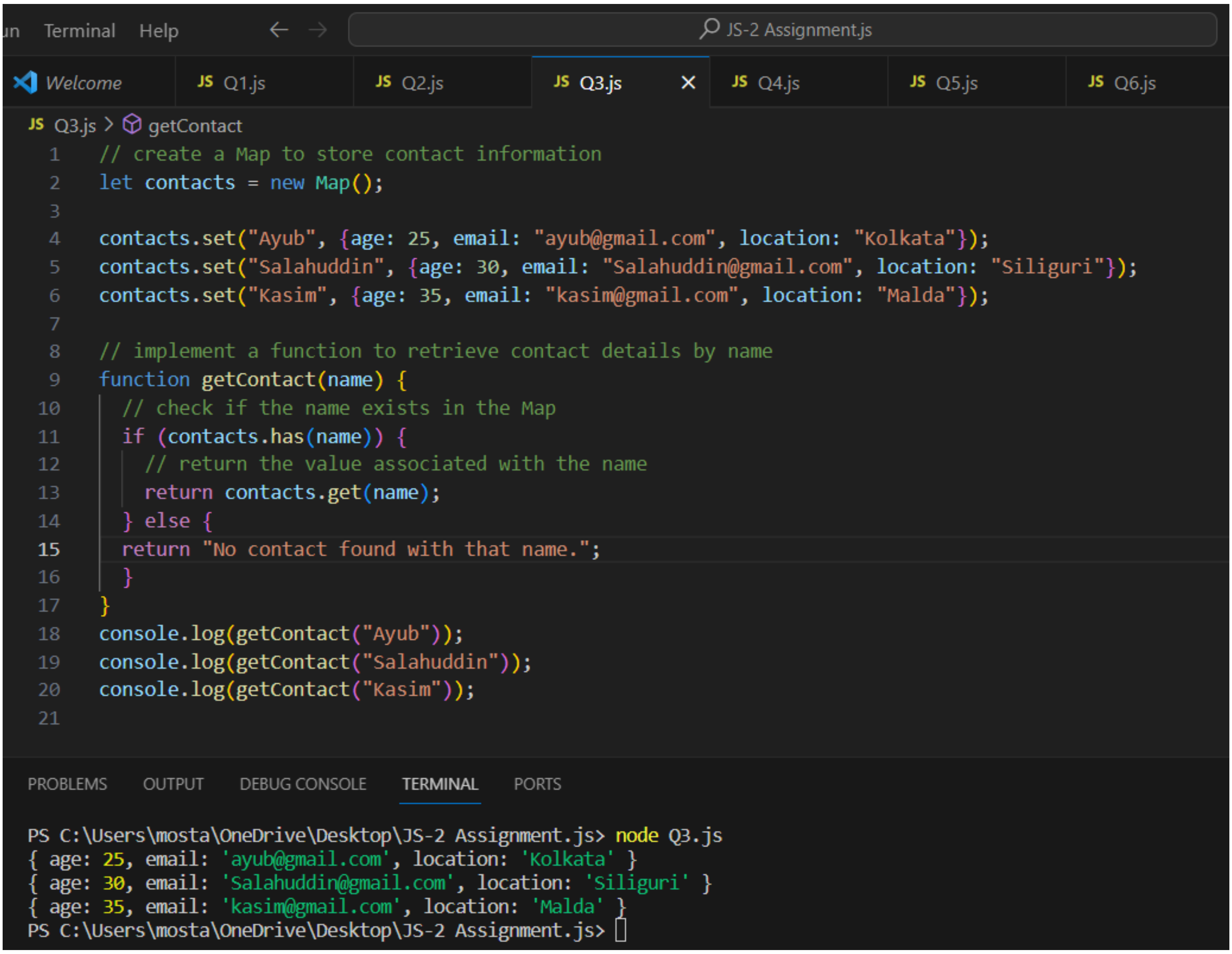
```
const ages=[19,22,19,24,20,25,26,24,25,24]
console.log(ages.sort())  // sorting
let min = ages.reduce((acc, cur) => Math.min(acc, cur)); // returns the minimum element of ages
let max = ages.reduce((acc, cur) => Math.max(acc, cur)); // returns the maximum element of ages
console.log("The minimum age is",min);
console.log("The maximum is",max);
function medianAge(ages) {
  let length = ages.length;
  if (length % 2 == 1) {
    return ages[length / 2];
  } else {
    return (ages[length / 2 - 1] + ages[length / 2]) / 2;
  }
}
console.log("The median age is",medianAge(ages));   // returns the median age of the ages
function averageAge(ages) {
  let sum = ages.reduce((acc, cur) => acc + cur); // add up all the elements
  return sum / ages.length; // divide by the number of elements
}
console.log("The average age is",averageAge(ages))  // returns the average age os the ages
function rangeAge(ages){
    return max-min
}
console.log("The range of the ages is",rangeAge(ages))  // returns the range of the ages
function compareDiff(min, max, average) {
  let maxDiff = Math.abs(max - average);
  let minDiff = Math.abs(min - average);
  return [maxDiff, minDiff];
}
console.log(compareDiff(19, 26, 22.8));
```



```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q2.js
[
  19, 19, 20, 22, 24,
  24, 24, 25, 25, 26
]
The minimum age is 19
The maximum is 26
The median age is 24
The average age is 22.8
The range of the ages is 7
[ 3.1999999999999993, 3.8000000000000007 ]
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> 
```

3. Create a Map to store contact information (name, age, email, location) and implement a function to retrieve contact details by name.

Solution:-

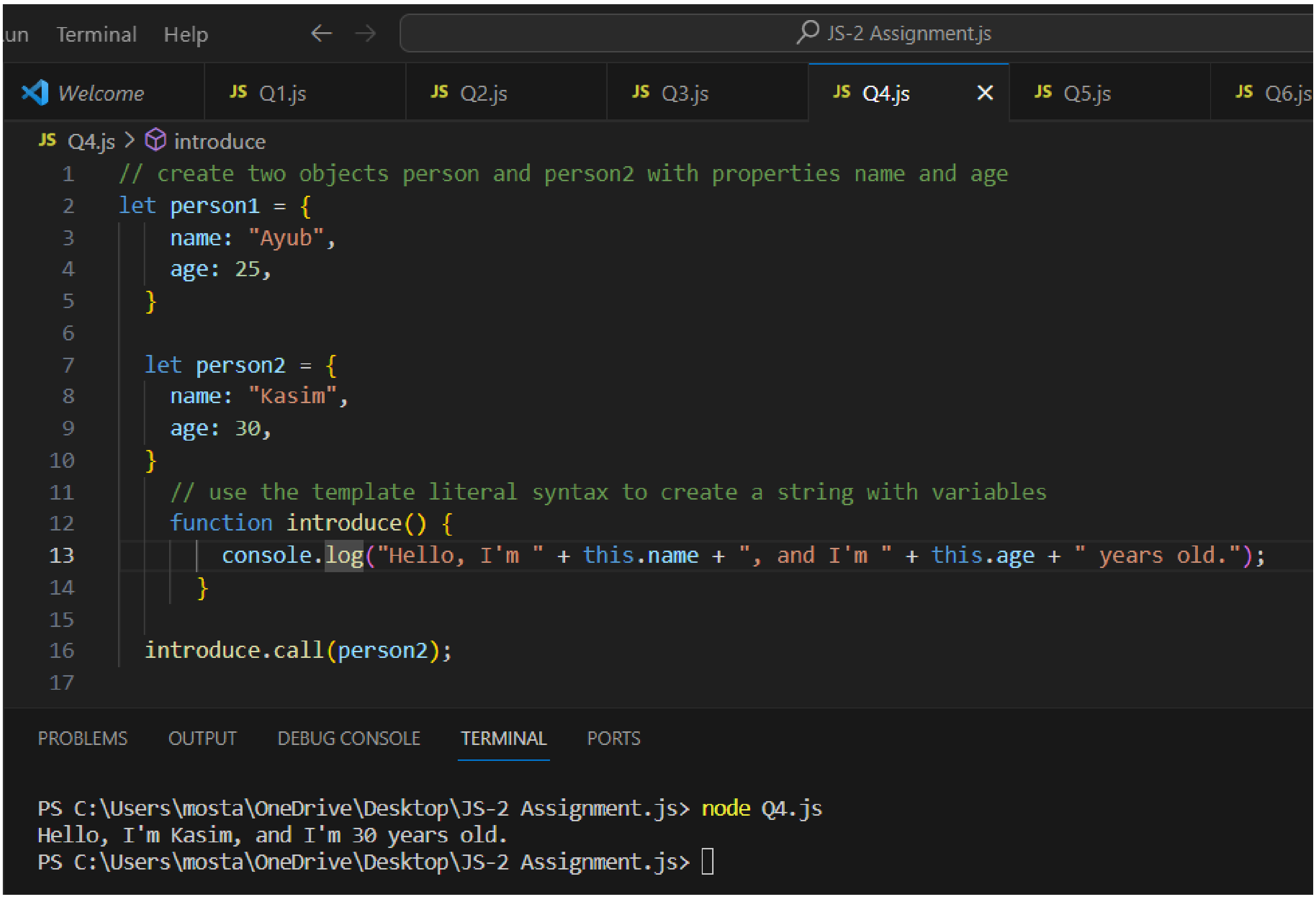


```javascript
// create a Map to store contact information
let contacts = new Map();

contacts.set("Ayub", {age: 25, email: "ayub@gmail.com", location: "Kolkata"});
contacts.set("Salahuddin", {age: 30, email: "Salahuddin@gmail.com", location: "Siliguri"});
contacts.set("Kasim", {age: 35, email: "kasim@gmail.com", location: "Malda"});

// implement a function to retrieve contact details by name
function getContact(name) {
  // check if the name exists in the Map
  if (contacts.has(name)) {
    // return the value associated with the name
    return contacts.get(name);
  } else {
  return "No contact found with that name.";
  }
}
console.log(getContact("Ayub"));
console.log(getContact("Salahuddin"));
console.log(getContact("Kasim"));

```

```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q3.js
{ age: 25, email: 'ayub@gmail.com', location: 'Kolkata' }
{ age: 30, email: 'Salahuddin@gmail.com', location: 'Siliguri' }
{ age: 35, email: 'kasim@gmail.com', location: 'Malda' }
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> 
```

4. Create two objects person1 and person2 with properties name and age. Create a function "introduce" that prints "Hello, I'm [name], and I'm [age] years old." Use the call method to make person2 introduce itself using the introduce function.

Solution:-

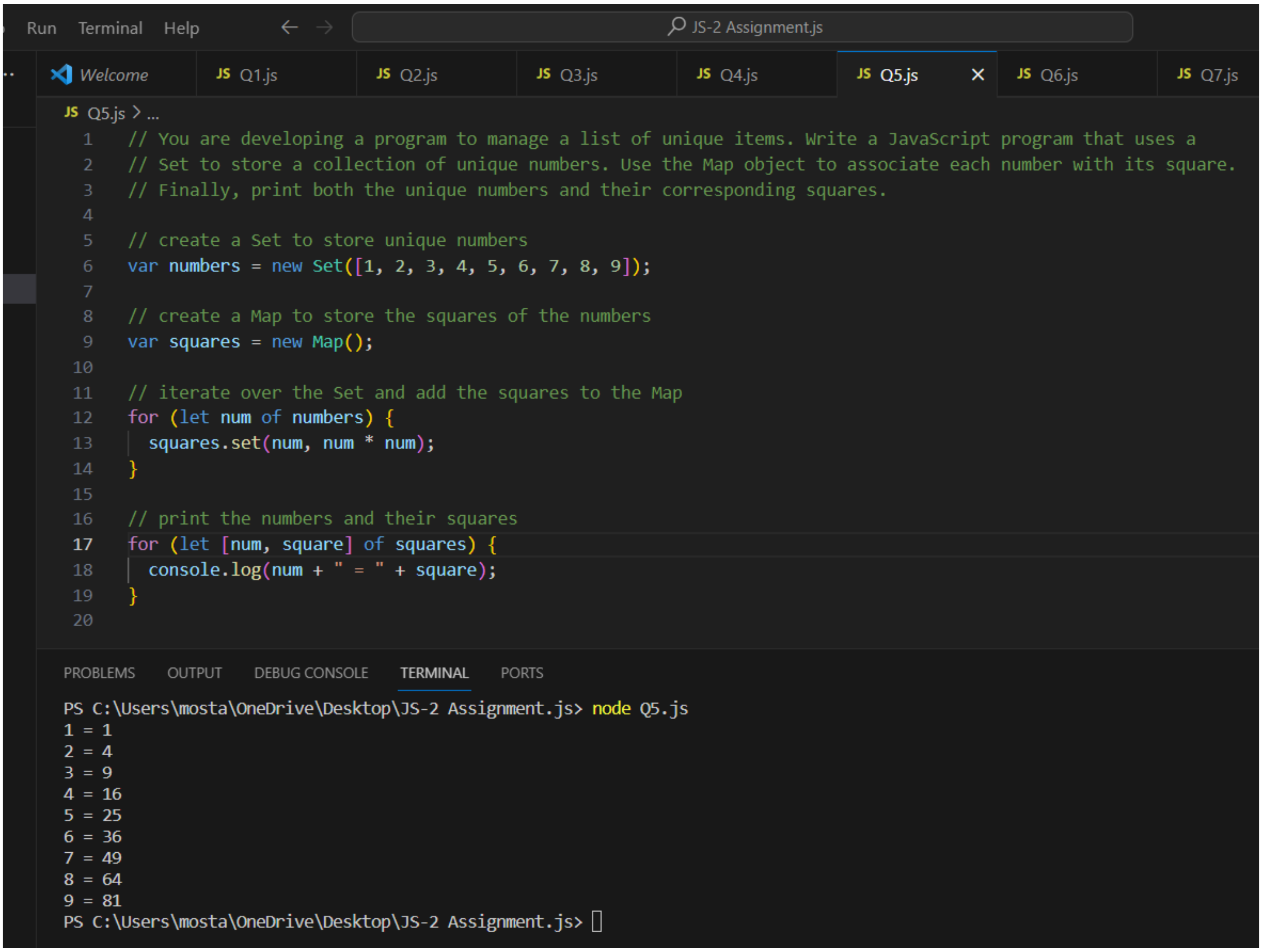


```javascript
// create two objects person and person2 with properties name and age
let person1 = {
    name: "Ayub",
    age: 25,
  }

  let person2 = {
    name: "Kasim",
    age: 30,
  }
    // use the template literal syntax to create a string with variables
    function introduce() {
        console.log("Hello, I'm " + this.name + ", and I'm " + this.age + " years old.");
      }

  introduce.call(person2);

```

```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q4.js
Hello, I'm Kasim, and I'm 30 years old.
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> 
```

5. You are developing a program to manage a list of unique items. Write a JavaScript program that uses a Set to store a collection of unique numbers. Use the Map object to associate each number with its square. Finally, print both the unique numbers and their corresponding squares.

Solution:-



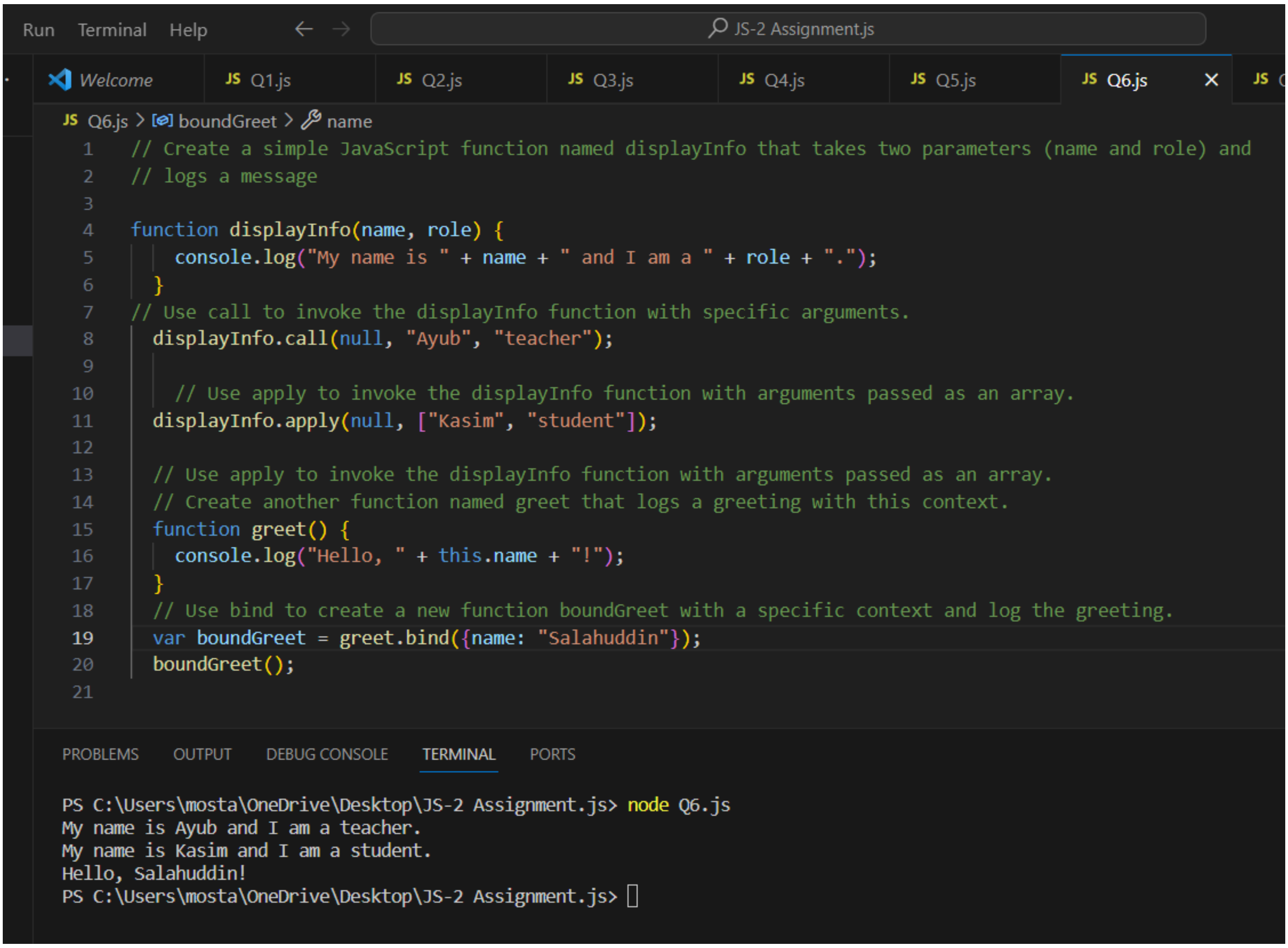
```javascript
// You are developing a program to manage a list of unique items. Write a JavaScript program that uses a
// Set to store a collection of unique numbers. Use the Map object to associate each number with its square.
// Finally, print both the unique numbers and their corresponding squares.

// create a Set to store unique numbers
var numbers = new Set([1, 2, 3, 4, 5, 6, 7, 8, 9]);

// create a Map to store the squares of the numbers
var squares = new Map();

// iterate over the Set and add the squares to the Map
for (let num of numbers) {
  squares.set(num, num * num);
}

// print the numbers and their squares
for (let [num, square] of squares) {
  console.log(num + " = " + square);
}
```

```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q5.js
1 = 1
2 = 4
3 = 9
4 = 16
5 = 25
6 = 36
7 = 49
8 = 64
9 = 81
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> 
```

6

- **Create a simple JavaScript function named displayInfo that takes two parameters (name and role) and logs a message.**
- **Use call to invoke the displayInfo function with specific arguments.**
- **Use apply to invoke the displayInfo function with arguments passed as an array.**
- **Create another function named greet that logs a greeting with this context.**
- **Use bind to create a new function boundGreet with a specific context and log the greeting.**

Solution:-



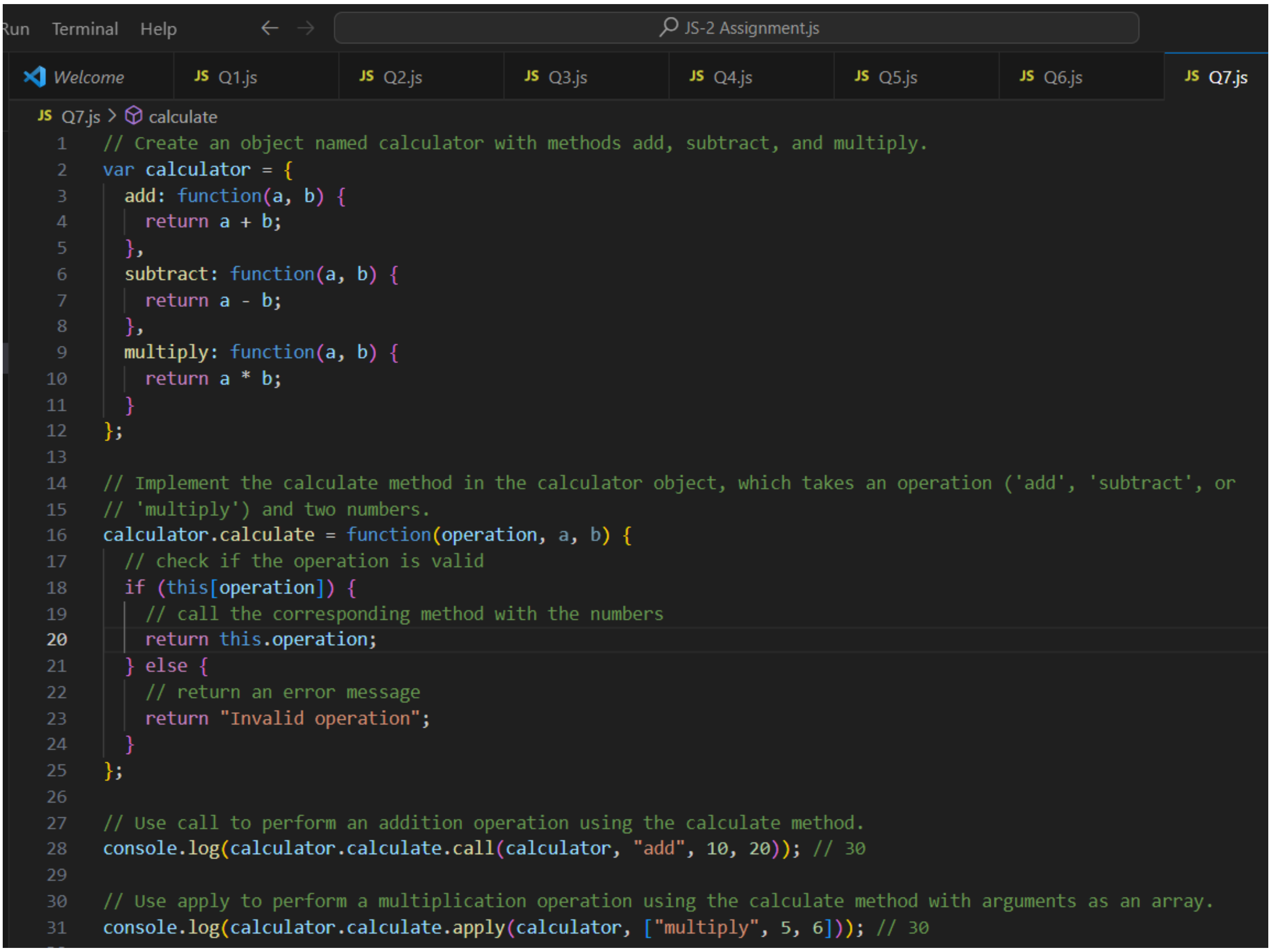
```javascript
// Create a simple JavaScript function named displayInfo that takes two parameters (name and role) and
// logs a message

function displayInfo(name, role) {
    console.log("My name is " + name + " and I am a " + role + ".");
  }
// Use call to invoke the displayInfo function with specific arguments.
  displayInfo.call(null, "Ayub", "teacher");

    // Use apply to invoke the displayInfo function with arguments passed as an array.
  displayInfo.apply(null, ["Kasim", "student"]);

  // Use apply to invoke the displayInfo function with arguments passed as an array.
  // Create another function named greet that logs a greeting with this context.
  function greet() {
    console.log("Hello, " + this.name + "!");
  }
  // Use bind to create a new function boundGreet with a specific context and log the greeting.
  var boundGreet = greet.bind({name: "Salahuddin"});
  boundGreet();
```

```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q6.js
My name is Ayub and I am a teacher.
My name is Kasim and I am a student.
Hello, Salahuddin!
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> 
```

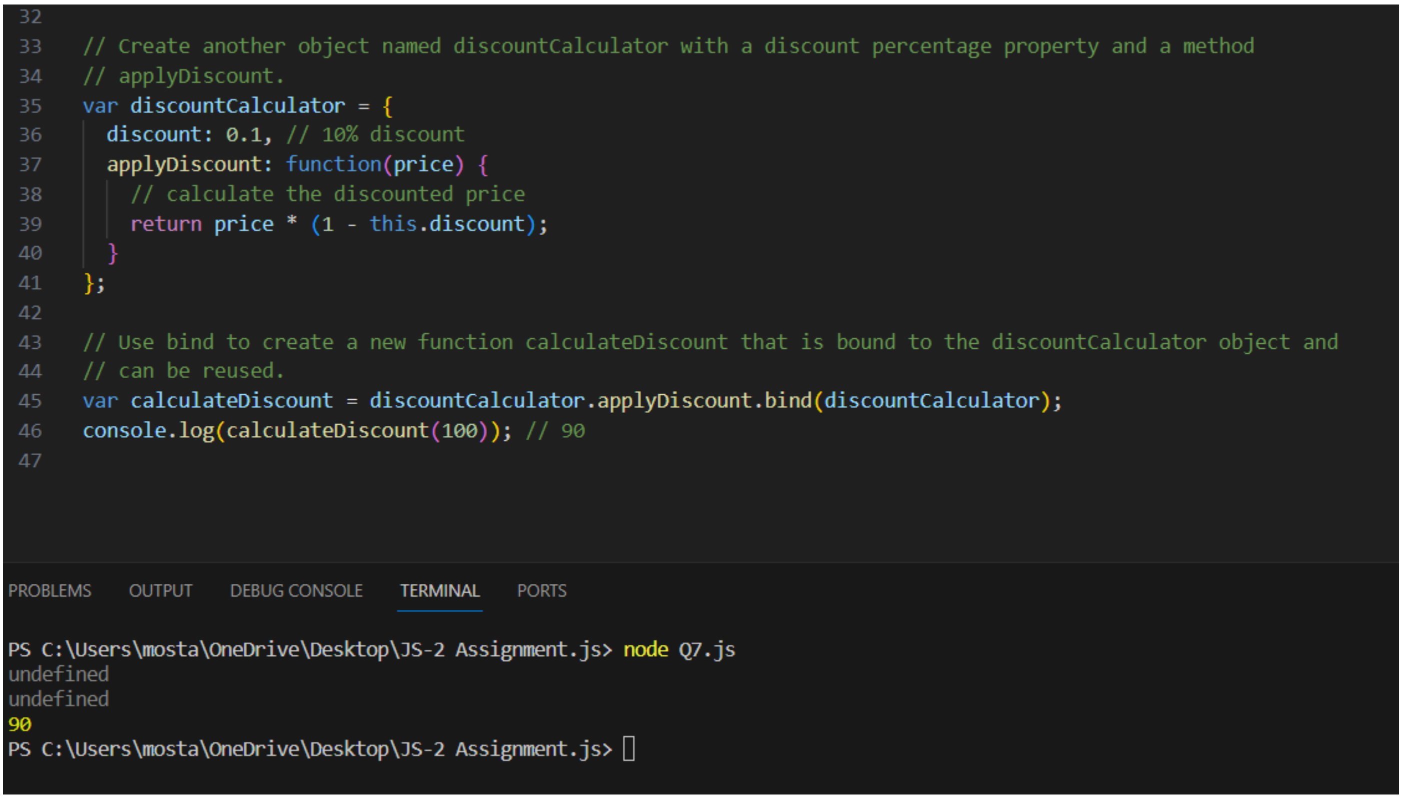
7. Tasks:

- **Create an object named calculator with methods add, subtract, and multiply.**
- **Implement the calculate method in the calculator object, which takes an operation ('add', 'subtract', or 'multiply') and two numbers.**
- **Use call to perform an addition operation using the calculate method.**
- **Use apply to perform a multiplication operation using the calculate method with arguments as an array.**
- **Create another object named discountCalculator with a discount percentage property and a method applyDiscount.**
- **Use bind to create a new function calculateDiscount that is bound to the discountCalculator object and can be reused.**

Solution:-



```
// Create an object named calculator with methods add, subtract, and multiply.
var calculator = {
  add: function(a, b) {
    return a + b;
  },
  subtract: function(a, b) {
    return a - b;
  },
  multiply: function(a, b) {
    return a * b;
  }
};

// Implement the calculate method in the calculator object, which takes an operation ('add', 'subtract', or
// 'multiply') and two numbers.
calculator.calculate = function(operation, a, b) {
  // check if the operation is valid
  if (this[operation]) {
    // call the corresponding method with the numbers
    return this.operation;
  } else {
    // return an error message
    return "Invalid operation";
  }
};

// Use call to perform an addition operation using the calculate method.
console.log(calculator.calculate.call(calculator, "add", 10, 20)); // 30

// Use apply to perform a multiplication operation using the calculate method with arguments as an array.
console.log(calculator.calculate.apply(calculator, ["multiply", 5, 6])); // 30
```



```javascript

// Create another object named discountCalculator with a discount percentage property and a method
// applyDiscount.
var discountCalculator = {
  discount: 0.1, // 10% discount
  applyDiscount: function(price) {
    // calculate the discounted price
    return price * (1 - this.discount);
  }
};

// Use bind to create a new function calculateDiscount that is bound to the discountCalculator object and
// can be reused.
var calculateDiscount = discountCalculator.applyDiscount.bind(discountCalculator);
console.log(calculateDiscount(100)); // 90

```

```
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> node Q7.js
undefined
undefined
90
PS C:\Users\mosta\OneDrive\Desktop\JS-2 Assignment.js> 
```