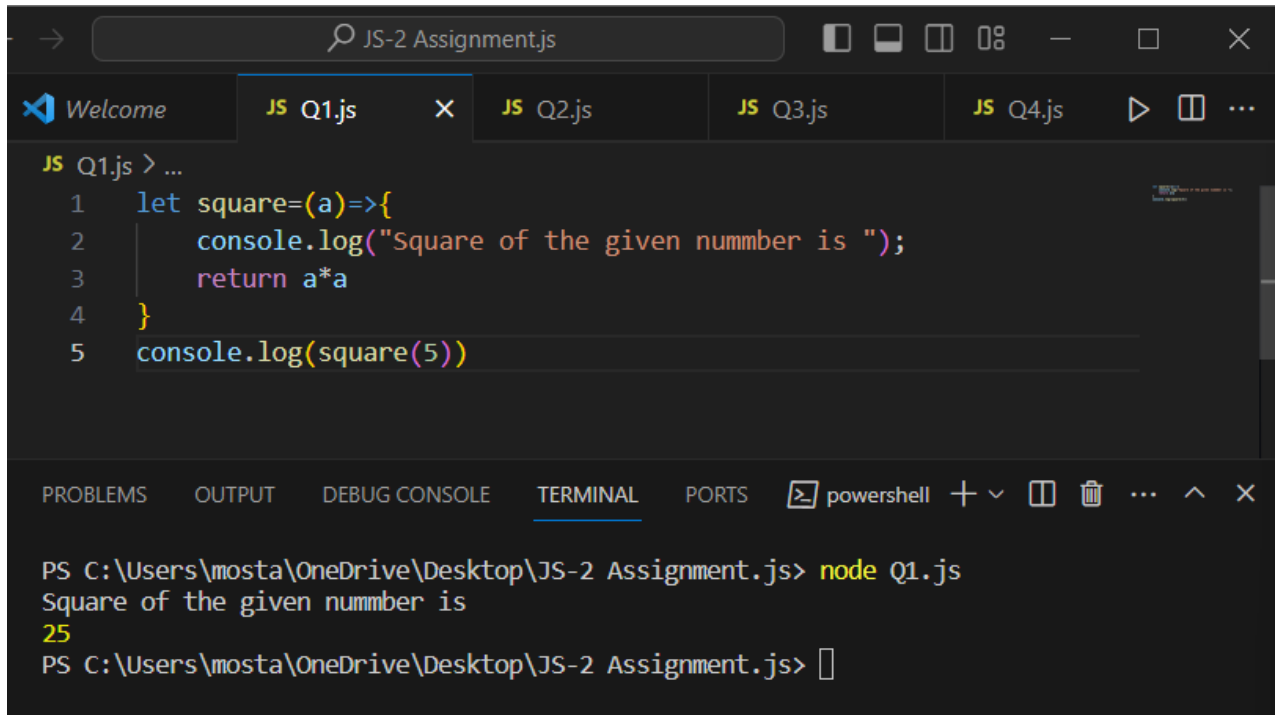


# Core Javascript-2

## 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:-



```
JS-2 Assignment.js
Welcome JS Q1.js JS Q2.js JS Q3.js JS Q4.js
JS Q1.js > ...
1 let square=(a)=>{
2   console.log("Square of the given nummber is ");
3   return a*a
4 }
5 console.log(square(5))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell
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:-

```
Terminal  Help  JS-2 Assignment.js

Welcome  JS Q1.js  JS Q2.js  JS Q3.js  JS Q4.js  JS Q5.js  JS Q6.js  JS

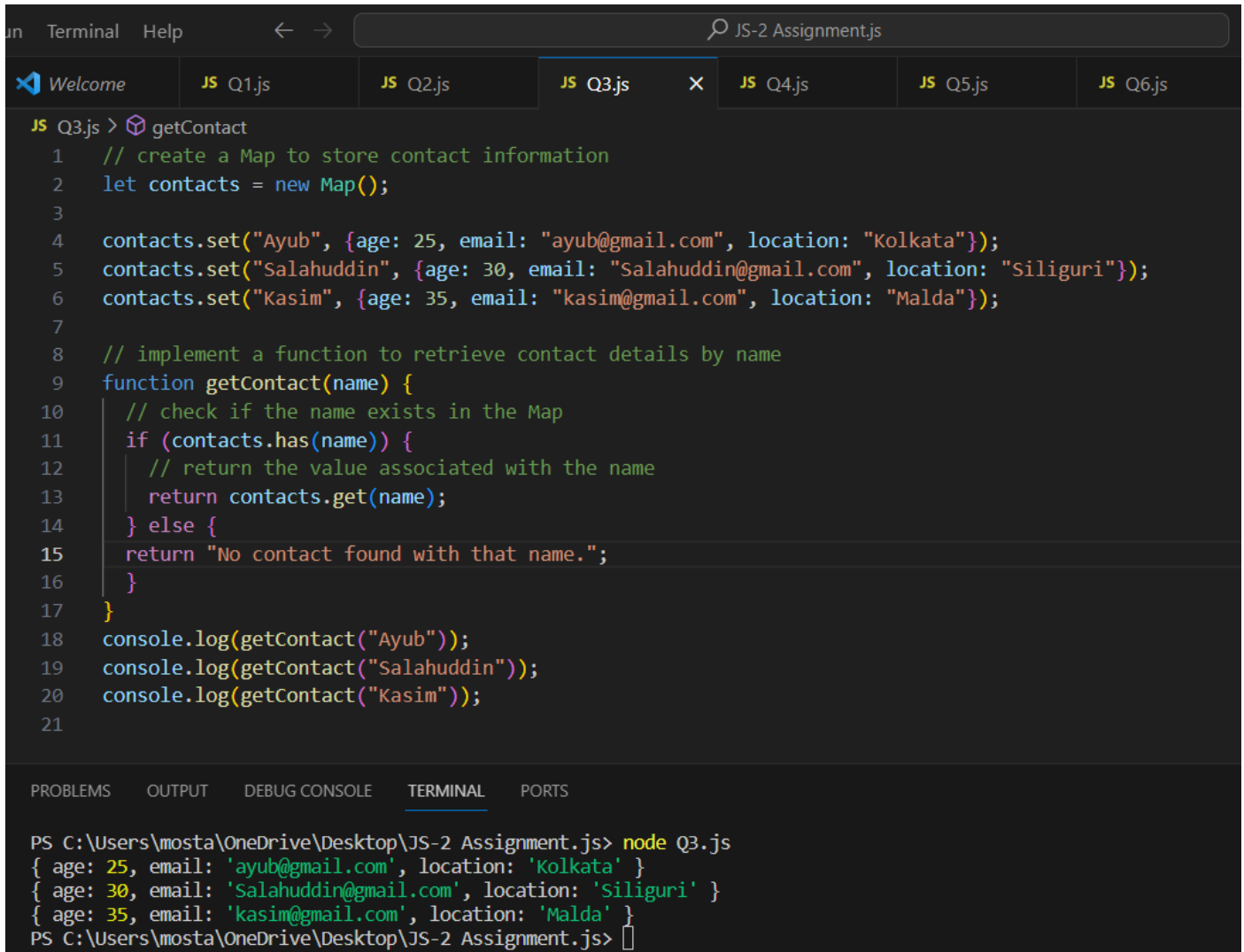
$ Q2.js > rangeAge
1  const ages=[19,22,19,24,20,25,26,24,25,24]
2  console.log(ages.sort()) // sorting
3  let min = ages.reduce((acc, cur) => Math.min(acc, cur)); // returns the minimum element of ages
4  let max = ages.reduce((acc, cur) => Math.max(acc, cur)); // returns the maximum element of ages
5  console.log("The minimum age is",min);
6  console.log("The maximum is",max);
7  function medianAge(ages) {
8      let length = ages.length;
9      if (length % 2 == 1) {
10         return ages[length / 2];
11     } else {
12         return (ages[length / 2 - 1] + ages[length / 2]) / 2;
13     }
14 }
15 console.log("The median age is",medianAge(ages)); // returns the median age of the ages
16 function averageAge(ages) {
17     let sum = ages.reduce((acc, cur) => acc + cur); // add up all the elements
18     return sum / ages.length; // divide by the number of elements
19 }
20 console.log("The average age is",averageAge(ages)) // returns the average age os the ages
21 function rangeAge(ages){
22     return max-min
23 }
24 console.log("The range of the ages is",rangeAge(ages)) // returns the range of the ages
25 function compareDiff(min, max, average) {
26     let maxDiff = Math.abs(max - average);
27     let minDiff = Math.abs(min - average);
28     return [maxDiff, minDiff];
29 }
30 console.log(compareDiff(19, 26, 22.8));
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

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:-



The image shows a Visual Studio Code editor window with a file named "JS-2 Assignment.js" open. The editor has several tabs, with "JS Q3.js" selected. The code in the file is as follows:

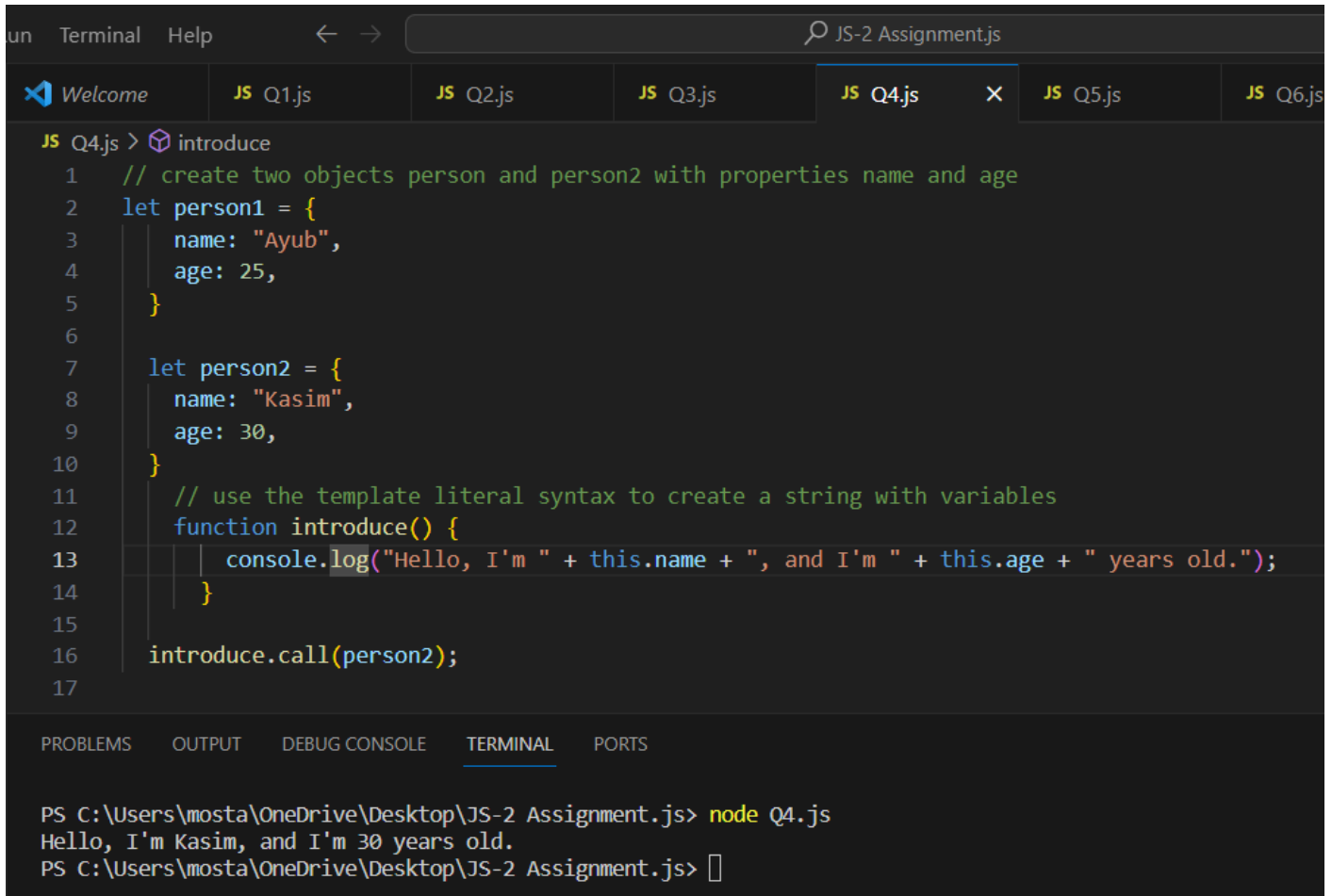
```
JS Q3.js > getContact
1 // create a Map to store contact information
2 let contacts = new Map();
3
4 contacts.set("Ayub", {age: 25, email: "ayub@gmail.com", location: "Kolkata"});
5 contacts.set("Salahuddin", {age: 30, email: "Salahuddin@gmail.com", location: "Siliguri"});
6 contacts.set("Kasim", {age: 35, email: "kasim@gmail.com", location: "Malda"});
7
8 // implement a function to retrieve contact details by name
9 function getContact(name) {
10     // check if the name exists in the Map
11     if (contacts.has(name)) {
12         // return the value associated with the name
13         return contacts.get(name);
14     } else {
15         return "No contact found with that name.";
16     }
17 }
18 console.log(getContact("Ayub"));
19 console.log(getContact("Salahuddin"));
20 console.log(getContact("Kasim"));
21
```

Below the editor, the terminal window shows the output of running the code with the command `node Q3.js`:

```
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:-



The image shows a Visual Studio Code editor window with a file named "JS-2 Assignment.js" open. The editor has several tabs at the top: "Welcome", "JS Q1.js", "JS Q2.js", "JS Q3.js", "JS Q4.js" (which is the active tab), "JS Q5.js", and "JS Q6.js". The code in "JS Q4.js" is as follows:

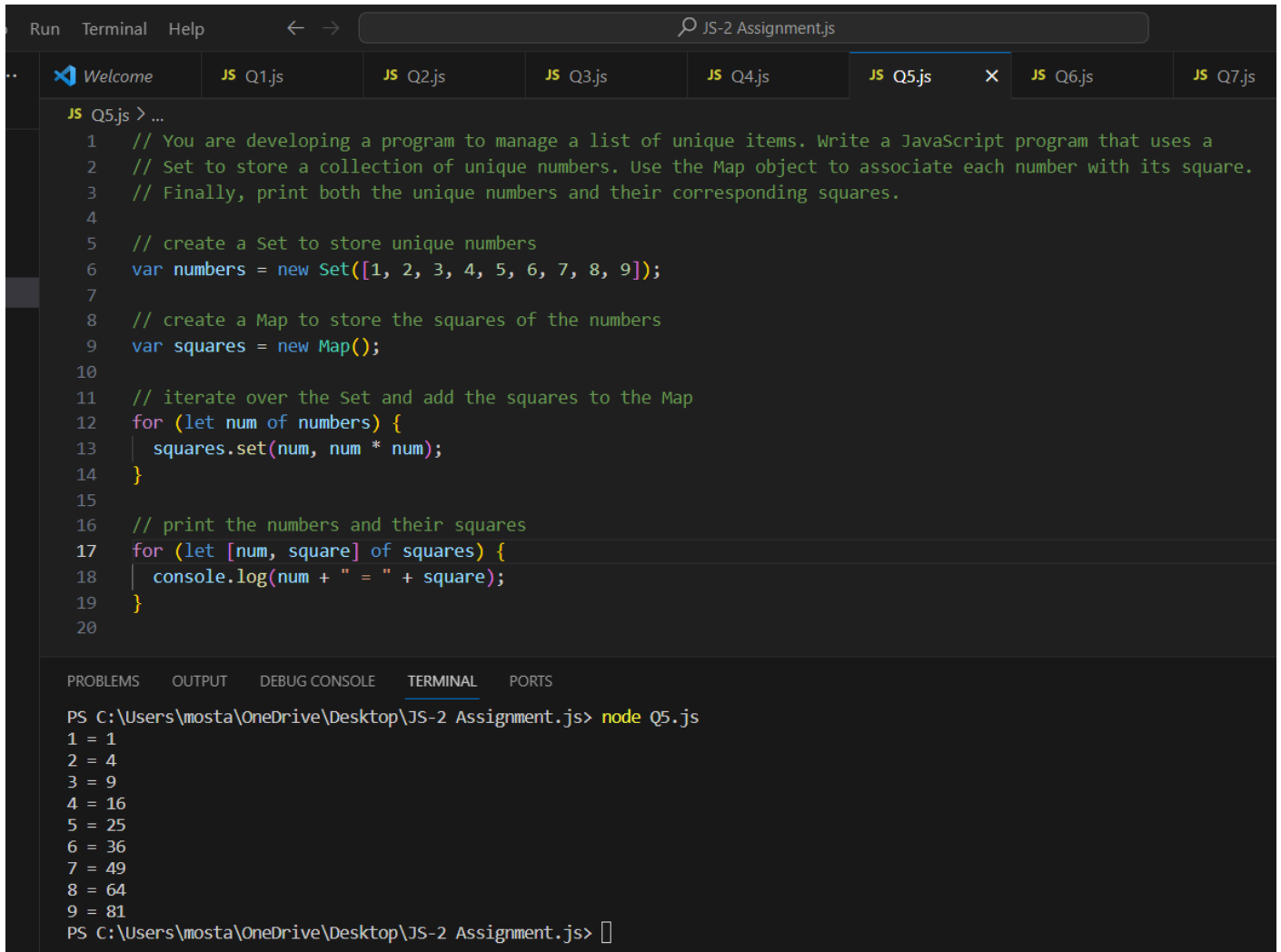
```
JS Q4.js > introduce
1 // create two objects person and person2 with properties name and age
2 let person1 = {
3     name: "Ayub",
4     age: 25,
5 }
6
7 let person2 = {
8     name: "Kasim",
9     age: 30,
10 }
11 // use the template literal syntax to create a string with variables
12 function introduce() {
13     console.log("Hello, I'm " + this.name + ", and I'm " + this.age + " years old.");
14 }
15
16 introduce.call(person2);
17
```

Below the editor, the "TERMINAL" tab is active, showing the command prompt output:

```
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:-



The screenshot shows the Visual Studio Code editor with a file explorer at the top displaying several JavaScript files (Q1.js to Q7.js). The editor window shows the code for Q5.js, which implements the solution to the problem. The code uses a Set to store unique numbers and a Map to store their squares. The terminal at the bottom shows the command to run the file and the resulting output.

```
JS Q5.js > ...
1 // You are developing a program to manage a list of unique items. Write a JavaScript program that uses a
2 // Set to store a collection of unique numbers. Use the Map object to associate each number with its square.
3 // Finally, print both the unique numbers and their corresponding squares.
4
5 // create a Set to store unique numbers
6 var numbers = new Set([1, 2, 3, 4, 5, 6, 7, 8, 9]);
7
8 // create a Map to store the squares of the numbers
9 var squares = new Map();
10
11 // iterate over the Set and add the squares to the Map
12 for (let num of numbers) {
13   squares.set(num, num * num);
14 }
15
16 // print the numbers and their squares
17 for (let [num, square] of squares) {
18   console.log(num + " = " + square);
19 }
20
```

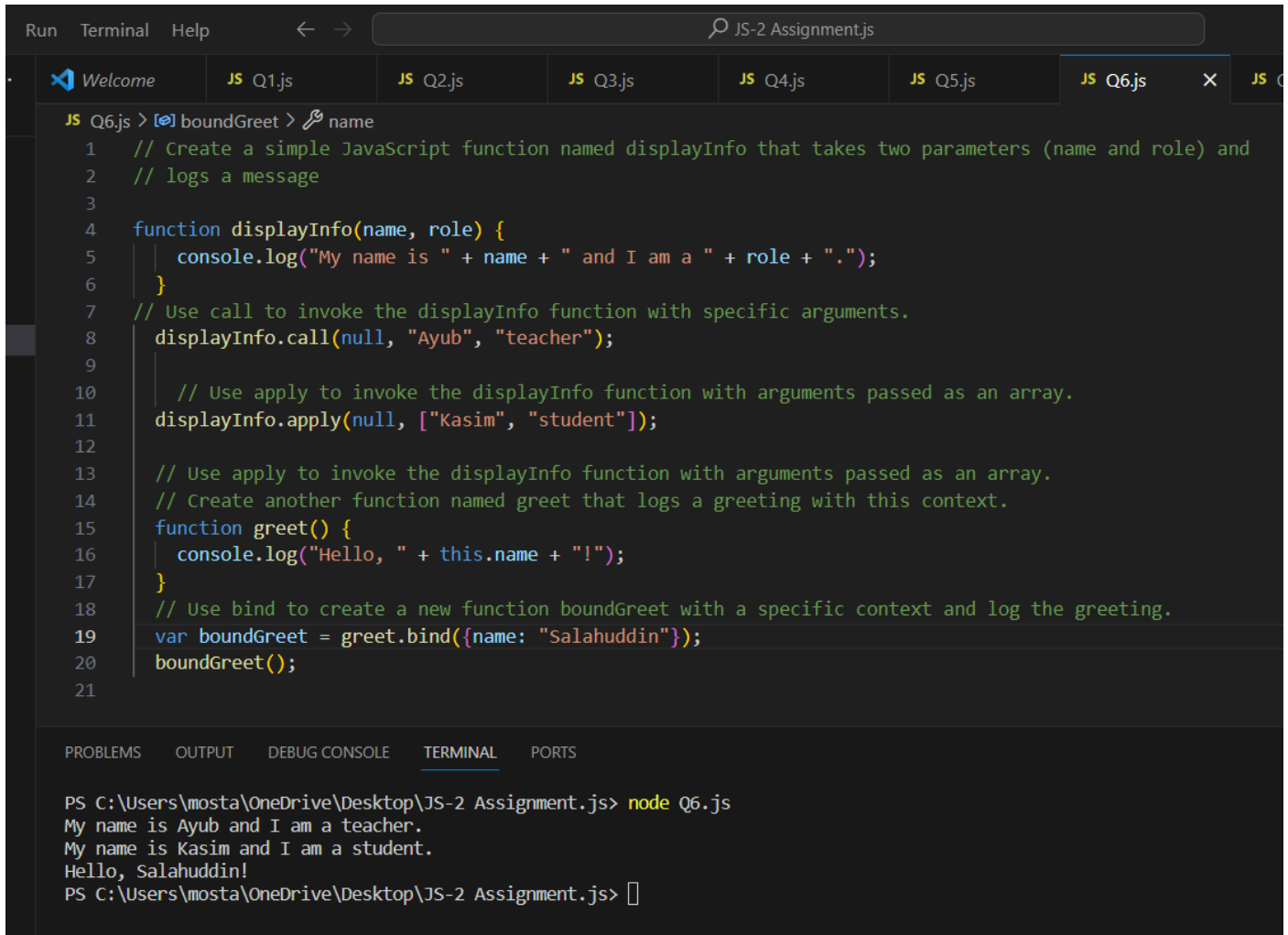
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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:-



```
JS Q6.js > boundGreet > name
1 // Create a simple JavaScript function named displayInfo that takes two parameters (name and role) and
2 // logs a message
3
4 function displayInfo(name, role) {
5     console.log("My name is " + name + " and I am a " + role + ".");
6 }
7 // Use call to invoke the displayInfo function with specific arguments.
8 displayInfo.call(null, "Ayub", "teacher");
9
10 // Use apply to invoke the displayInfo function with arguments passed as an array.
11 displayInfo.apply(null, ["Kasim", "student"]);
12
13 // Use apply to invoke the displayInfo function with arguments passed as an array.
14 // Create another function named greet that logs a greeting with this context.
15 function greet() {
16     console.log("Hello, " + this.name + "!");
17 }
18 // Use bind to create a new function boundGreet with a specific context and log the greeting.
19 var boundGreet = greet.bind({name: "Salahuddin"});
20 boundGreet();
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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:-

```
Run Terminal Help JS-2 Assignment.js
Welcome JS Q1.js JS Q2.js JS Q3.js JS Q4.js JS Q5.js JS Q6.js JS Q7.js
JS Q7.js > calculate
1 // Create an object named calculator with methods add, subtract, and multiply.
2 var calculator = {
3   add: function(a, b) {
4     return a + b;
5   },
6   subtract: function(a, b) {
7     return a - b;
8   },
9   multiply: function(a, b) {
10    return a * b;
11  }
12 };
13
14 // Implement the calculate method in the calculator object, which takes an operation ('add', 'subtract', or
15 // 'multiply') and two numbers.
16 calculator.calculate = function(operation, a, b) {
17   // check if the operation is valid
18   if (this[operation]) {
19     // call the corresponding method with the numbers
20     return this[operation](a, b);
21   } else {
22     // return an error message
23     return "Invalid operation";
24   }
25 };
26
27 // Use call to perform an addition operation using the calculate method.
28 console.log(calculator.calculate.call(calculator, "add", 10, 20)); // 30
29
30 // Use apply to perform a multiplication operation using the calculate method with arguments as an array.
31 console.log(calculator.calculate.apply(calculator, ["multiply", 5, 6])); // 30
```

```
32
33 // Create another object named discountCalculator with a discount percentage property and a method
34 // applyDiscount.
35 var discountCalculator = {
36     discount: 0.1, // 10% discount
37     applyDiscount: function(price) {
38         // calculate the discounted price
39         return price * (1 - this.discount);
40     }
41 };
42
43 // Use bind to create a new function calculateDiscount that is bound to the discountCalculator object and
44 // can be reused.
45 var calculateDiscount = discountCalculator.applyDiscount.bind(discountCalculator);
46 console.log(calculateDiscount(100)); // 90
47
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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> 
```