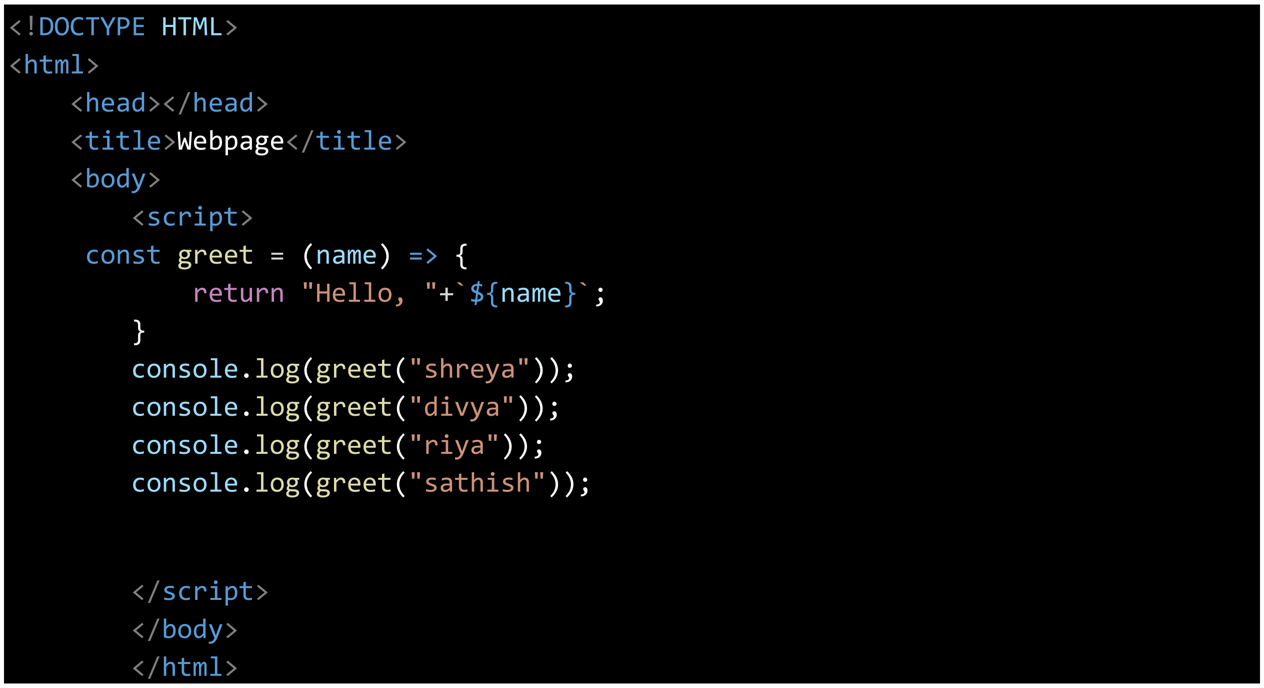
NAME:P.V.SHREYA SREE

CLASS:EEE-B

ROLL NO:717823E253

51.

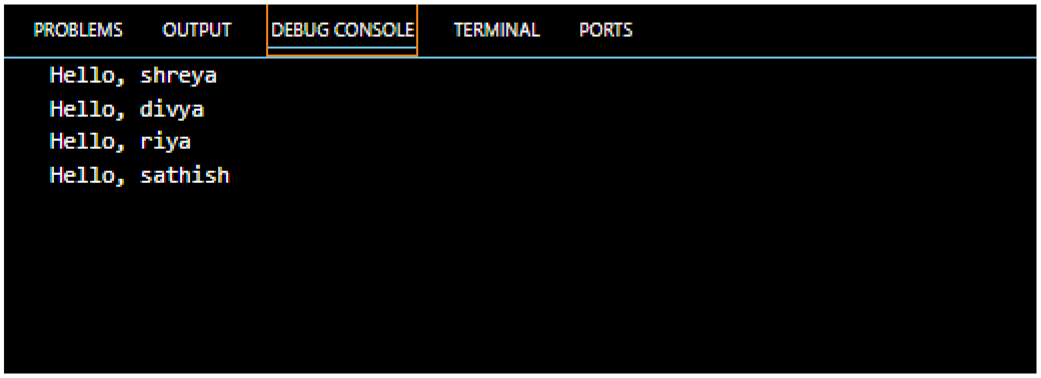
Task 51: Declare a simple arrow function named greet that takes one parameter name and returns the string “Hello, name!”. Test your function with various names.



```
<!DOCTYPE HTML>
<html>
    <head></head>
    <title>Webpage</title>
    <body>
        <script>
     const greet = (name) => {
            return "Hello, "+`${name}`;
        }
        console.log(greet("shreya"));
        console.log(greet("divya"));
        console.log(greet("riya"));
        console.log(greet("sathish"));


        </script>
        </body>
        </html>
```

OUTPUT:



```
Hello, shreya
Hello, divya
Hello, riya
Hello, sathish
```

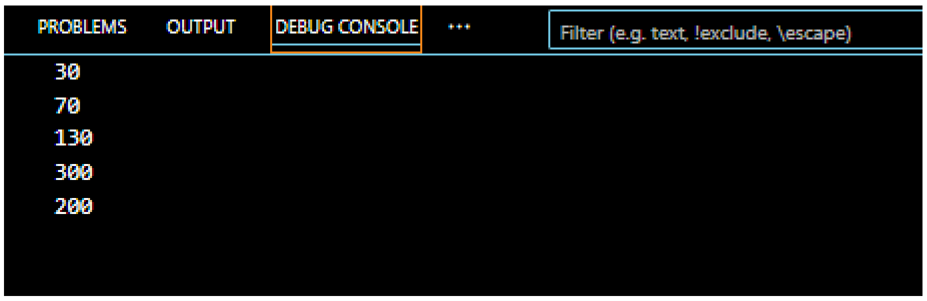
52.

Task 52: Write an arrow function named add that takes two parameters and returns their sum. Validate your function with several pairs of numbers.

```
<!DOCTYPE HTML>
<html>
    <head></head>
    <title>Webpage</title>
    <body>
        <script>
     const add = (num1,num2) => {
            return num1+num2;
        }
        console.log(add(10,20));
        console.log(add(30,40));
        console.log(add(60,70));
        console.log(add(100,200));
        console.log(add(100,100));

        </script>
        </body>
        </html>
```

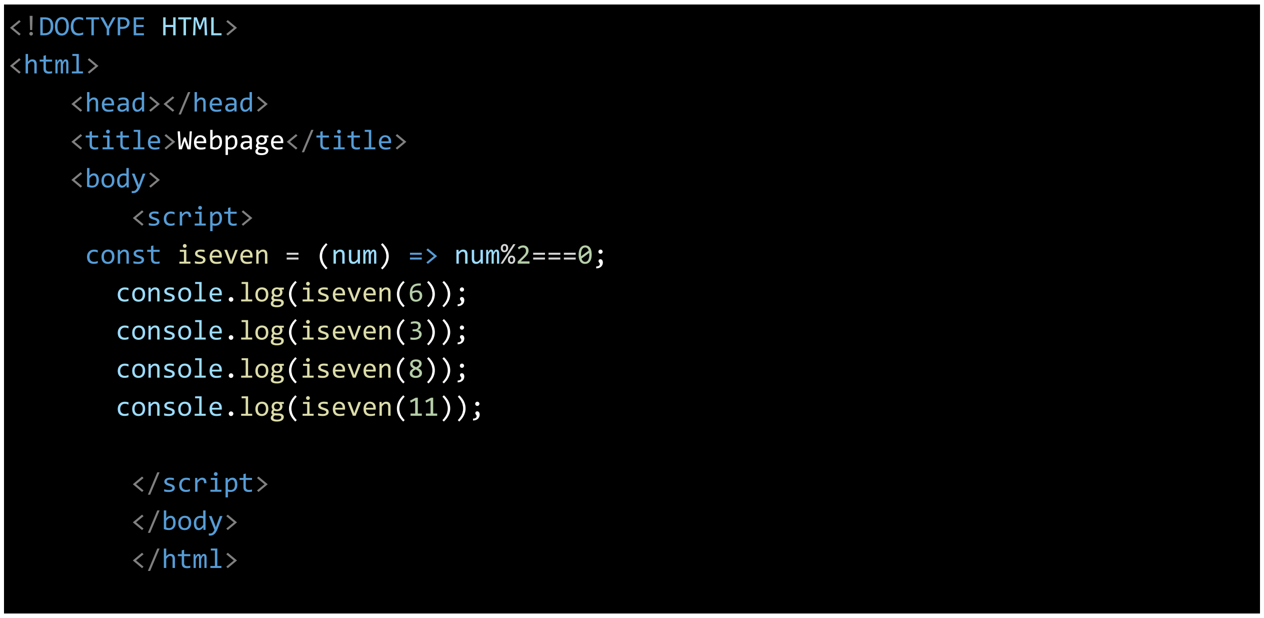
OUTPUT:



```
30
70
130
300
200
```

53.

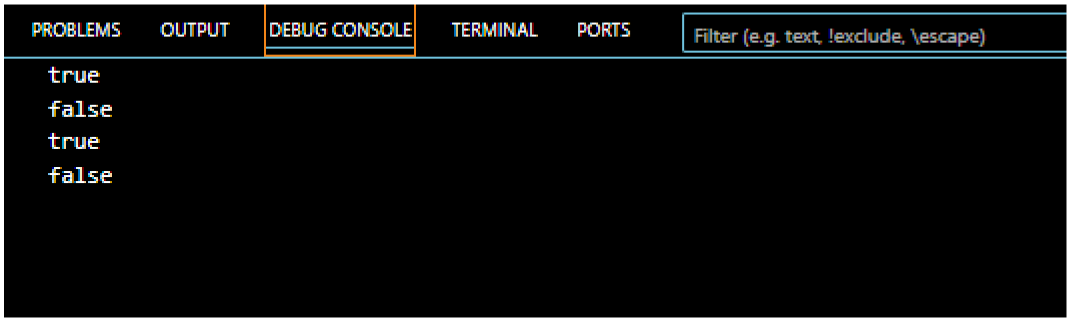
Task 53: Declare an arrow function named isEven that checks if a number is even. If the number is even, it should return true; otherwise, false. Remember that if the arrow function body has a single statement, you can omit the curly braces



```html
<!DOCTYPE HTML>
<html>
    <head></head>
    <title>Webpage</title>
    <body>
        <script>
     const iseven = (num) => num%2===0;
       console.log(iseven(6));
       console.log(iseven(3));
       console.log(iseven(8));
       console.log(iseven(11));

        </script>
        </body>
        </html>
```

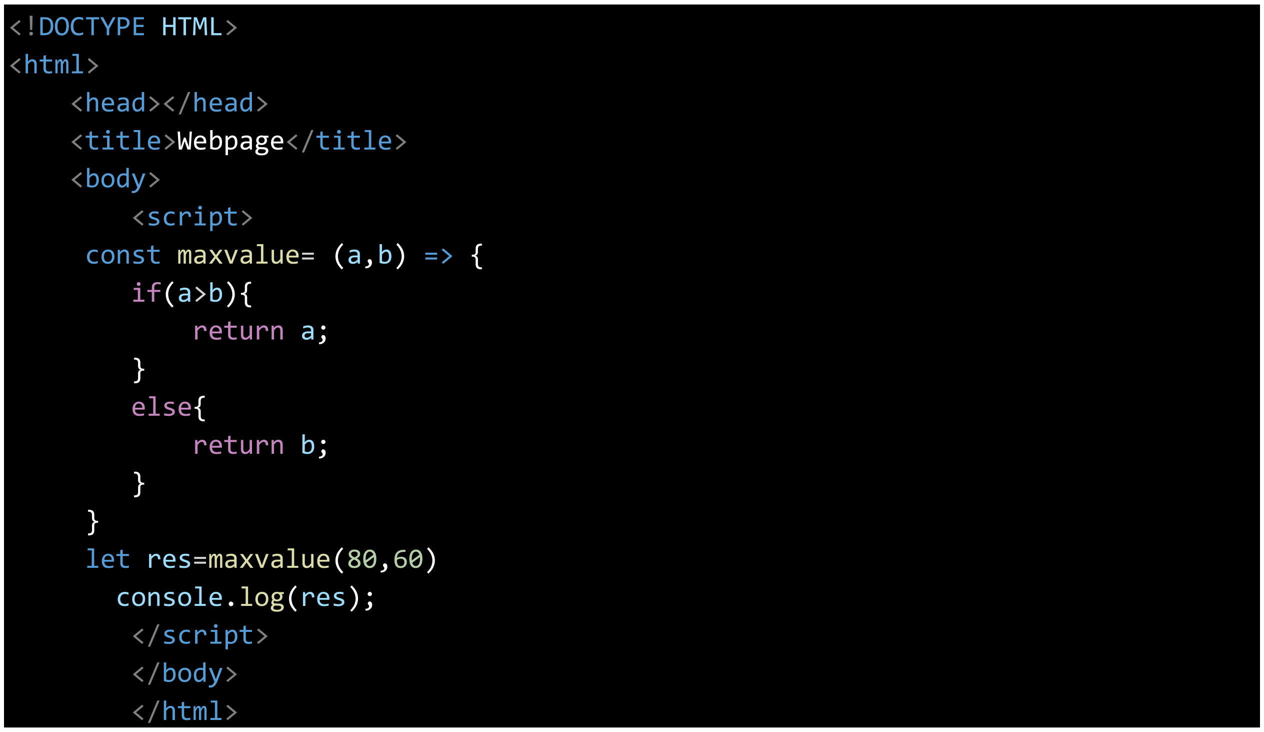
OUTPUT:



```
true
false
true
false
```

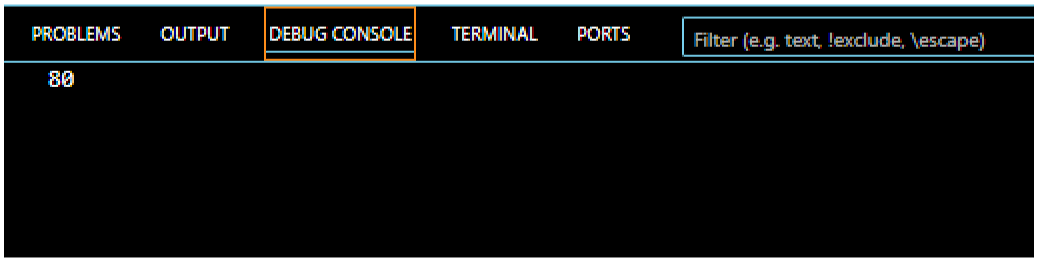
54.

Task 54: Implement an arrow function named maxValue that takes two numbers as parameters and returns the larger number. Here, you'll need to use curly braces for the function body and the return statement.



```
<!DOCTYPE HTML>
<html>
    <head></head>
    <title>Webpage</title>
    <body>
        <script>
     const maxvalue= (a,b) => {
        if(a>b){
            return a;
        }
        else{
            return b;
        }
     }
     let res=maxvalue(80,60)
       console.log(res);
        </script>
        </body>
        </html>
```

OUTPUT:



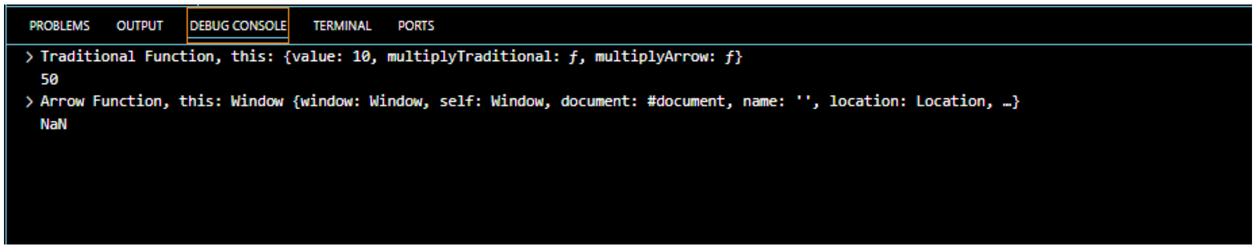
```
80
```

55.

Task 55: Examine the behavior of the this keyword inside an arrow function vs a traditional function. Create an object named myObject with a property value set to 10 and two methods: multiplyTraditional using a traditional function and multiplyArrow using an arrow function. Both methods should attempt to multiply the value property by a number passed as a parameter. Check the value of this inside both methods

```html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Task 55: Behavior of 'this' in Arrow vs Traditional
Function</title>
    </head>
    <body>
        <script>
            const myObject = {
                value: 10,
                multiplyTraditional: function(num) {
                    console.log("Traditional Function, this:", this);
                    return this.value * num;
                },
                multiplyArrow: (num) => {
                    console.log("Arrow Function, this:", this);
                    return this.value * num;
                }
            };
            console.log(myObject.multiplyTraditional(5));
            console.log(myObject.multiplyArrow(5));
        </script>
    </body>
</html>
```

OUTPUT:



```
> Traditional Function, this: {value: 10, multiplyTraditional: ƒ, multiplyArrow: ƒ}
  50
> Arrow Function, this: Window {window: Window, self: Window, document: #document, name: '', location: Location, …}
  NaN
```