Week 1

Exercise 1:

Output:

```
This is my first programme
Welcome John, your month's salary is £50,000
The sum of 5 and 3 is 8
```

Exercise 2:

Output:

```
Enter a number: 99
This number is postitive
```

```
Enter a number: -2
This number is less than zero
```

```
Enter a number: 0
This number is zero
```

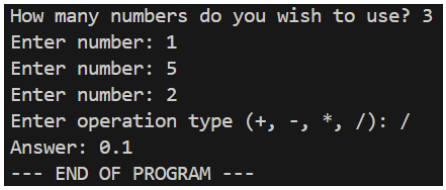
Exercise 3 – Calculator Program:

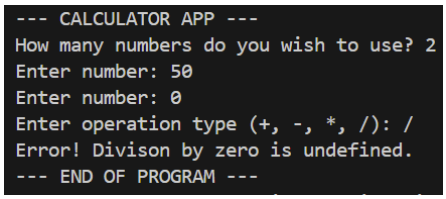
Output:

```
--- CALCULATOR APP ---
How many numbers do you wish to use? 5
Enter number: 3
Enter number: 6
Enter number: 1
Enter number: 50
Enter number: 41
Enter operation type (+, -, *, /): +
Answer: 101
--- END OF PROGRAM ---
```

```
--- CALCULATOR APP ---
How many numbers do you wish to use? 2
Enter number: 70
Enter number: 110
Enter operation type (+, -, *, /): -
Answer: -40
--- END OF PROGRAM ---
```

```
--- CALCULATOR APP ---
How many numbers do you wish to use? 4
Enter number: 3
Enter number: 10
Enter number: 7
Enter number: 100
Enter operation type (+, -, *, /): *
Answer: 21000
--- END OF PROGRAM ---
```





Week 1 Reflection

Exercise 1 taught me how to print text to a terminal using console.log, how to declare, initialize, and assign integer values to constants, and how to use basic addition. Exercise 2 taught me how to get integer input from a user using a prompt, and how to use conditionals – if, else if, and else.

For exercise 3, I decided to go beyond the basic requirements and create a calculator that can take as many operands as the user desires. I was able to achieve this with the use of an array. I learned that in JavaScript an array's length is not fixed and can thus take as many number of values as is needed. I was able to assign the user's input to each element space in the array using a for loop.

With the user choosing an operator, the best solution to applying said operator to the user's chosen numbers was with a switch statement. I had to pay special attention to each case as the initial answer value had to be different depending on the operator chosen. For example, the multiplicative case required that the first value was set to 1; and the subtraction case required that the first value was set to the first index value in the array. I also implemented an escape encase of the division-by-zero scenario.

Lastly, I was pleasantly surprised to find out that the basic syntax of JavaScript is similar to that of Java, which made it much easier to learn.