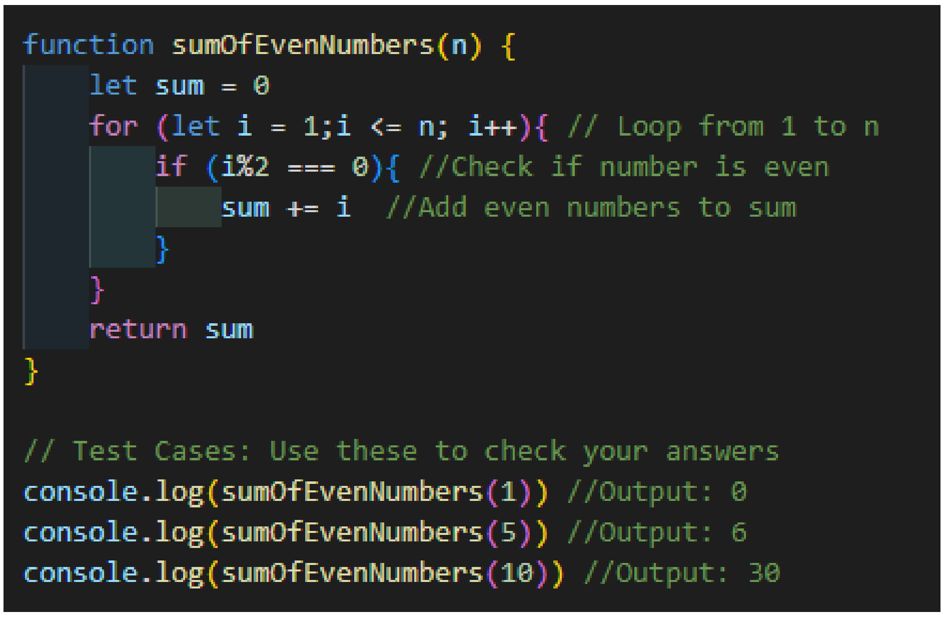
## Exercise 1:

Write a program that calculates the sum of all even numbers from 1 to a given number *n*.

Solution

- Use a Loop to Iterate from 1 to n
  - Use a *for loop* to iterate through each number from 1 to n.
- Check If the Number is Even
  - Inside the loop, use an if statement to check if the current number is even. *(if-else)*



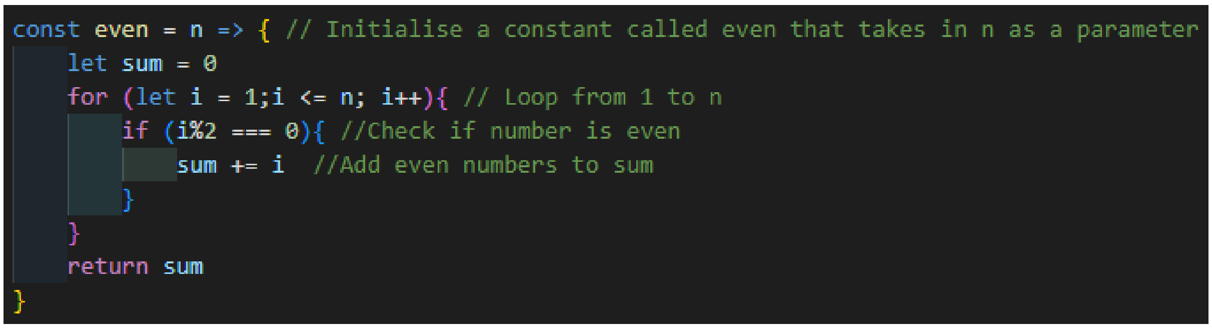
```
function sumOfEvenNumbers(n) {
    let sum = 0
    for (let i = 1;i <= n; i++){ // Loop from 1 to n
        if (i%2 === 0){ //Check if number is even
            sum += i  //Add even numbers to sum
        }
    }
    return sum
}

// Test Cases: Use these to check your answers
console.log(sumOfEvenNumbers(1)) //Output: 0
console.log(sumOfEvenNumbers(5)) //Output: 6
console.log(sumOfEvenNumbers(10)) //Output: 30
```

## Exercise 2:

Rewrite the code you did in exercise 1 using arrow function syntax and call it.

Solution



```
const even = n => { // Initialise a constant called even that takes in n as a parameter
    let sum = 0
    for (let i = 1;i <= n; i++){ // Loop from 1 to n
        if (i%2 === 0){ //Check if number is even
            sum += i  //Add even numbers to sum
        }
    }
    return sum
}
```

## Final Exercise

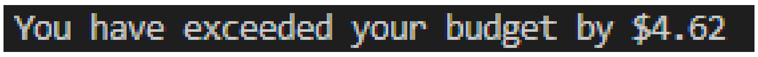
Write a program that manages a shopping cart. The list of items in the shopping cart and its price are shown below.

*Note: To round value to 2 decimal places, use:* `Math.round({value}*100)/100.`

| Item | Quantity | Price (for 1 quantity) |
|---|---|---|
| Milk | 2 | $3.49 |
| Apples | 6 | $1.80 |
| Oranges | 4 | $1.65 |
| Toothpaste | 2 | $4.00 |
| Detergent | 1 | $12.24 |

1. Jean has a budget of $40. Write a function that displays the total amount of items in her shopping cart. If the final total exceeds her budget, it should notify her and show how much she has exceeded by. *(Hint: Create another array for quantity purchased for easier calculation)*

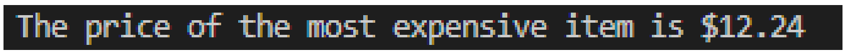
- Final Output:



```
You have exceeded your budget by $4.62
```

2. Write a program to determine the price of the most expensive item in her cart based on the total price for the item (calculated as unit price × quantity). For example:

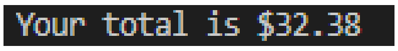
- If Jean buys 2 cartons of milk at $3.49 each, the total price for milk is $3.49 x 2 = $6.98.
- Final Output:



```
The price of the most expensive item is $12.24
```

3. Based on the output of qn2, Jean decides not to get the most expensive item.
Using the same file as qn1, without manually deleting an item from the array, remove the most expensive item from the cart and run your function again.

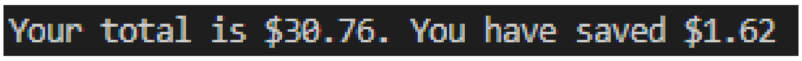
- Final Output:



```
Your total is $32.38
```

4. The supermarket is having a 5% discount on the final total. With the new cart, write a new function to calculate the total after discount and show how much Jean has saved.

- Final Output



```
Your total is $30.76. You have saved $1.62
```

## Extra Exercises:

### Extra 1: Temperature Tracker

You are creating a simple weather tracker that categorizes temperatures into "cold," "moderate," and "hot" based on a specific threshold.

**1a.** Write a function categoriseTemperature that accepts a single number (temperature in Celsius) and returns a string indicating the category of the temperature. *(Hint: You cannot chain comparisons (5 < i < 10) in JS, you will have to use logical operators.)*

- If the temperature is below 15°C, return "cold".
- If the temperature is between 15°C and 30°C (inclusive), return "moderate".
- If the temperature is above 30°C, return "hot".

**1b.** Modify your function in 1a so it accepts 2 parameters: 1) Current Temperature, 2) adjustment (can be positive or negative). The function should adjust and return the new temperature and its category after adjustment.

For example, if the function is: categoriseTemperature(10, 6), the function should output: 16°C, Moderate

**1c.** Write a function findHeatWave that accepts an array of temperatures for a week. It should identify if a heatwave has happened. A heatwave can only be considered to have happened if there are at least 3 consecutive days where the temperature is above 35°C.

If there is a heatwave, the function should return the day it started and the duration it lasted. If there are no heatwaves, return null.

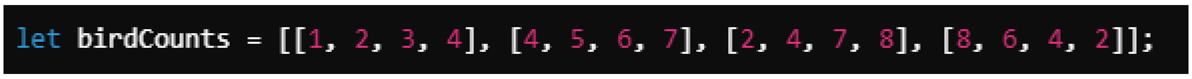
### Extra 2: Bird Watcher

You are an avid bird watcher that keeps track of how many birds have visited the nearby park. Usually, you use a tally in a notebook to count the birds but you want to digitalise your data to better work with it. The table below shows the number of birds you've seen today.

| **Type** | **Number seen** |
|---|---|
| Pigeon | 21 |
| Sparrow | 14 |
| Robin | 4 |
| Charmony Dove | 6 |
| Crow | 9 |
| Oriole | 5 |
| Koel | 3 |

**2a.** Implement a function totalBirdCount that accepts an array that contains the total bird count for the day. It should return the total number of birds that you've seen. *(Hint: You can look into* ***for…of*** *for array summing)*

**2b**. A month has passed since you started digitizing your birdwatching data. The bird counts for the past month are stored as an array, where each inner array represents a week, and each number inside the inner arrays represents the count for a specific day.



```
let birdCounts = [[1, 2, 3, 4], [4, 5, 6, 7], [2, 4, 7, 8], [8, 6, 4, 2]];
```

*For example: There are 4 inner arrays inside the outer array, each inner array represents 1 week. Inside each of these inner arrays are the total number of birds counted each day.*

Implement a function totalWeeklyCount that accepts an array and returns an array of the total number of birds seen each week.

**2c.** The park has become a popular spot for birdwatching, and you're curious to find out which week had the highest bird count in the past month. Implement a function busiestWeek that accepts an array similar to the one in question 2. The function should:

1. Identify the week with the highest bird count.
2. Return an string that contains:
   - The highest bird count.
   - The week (e.g., "Week 2") on which it occurred.
3. You can assume there won't be the same number of birds seen each week.

You can reuse your function in part 2b.