**BUỔI 01, 02: -JAVASCRIPT FUNDAMENTALS**

**Coding Challenge #1**

Mark and John are trying to compare their BMI (Body Mass Index), which is calculated using the formula:

BMI = mass / height \*\* 2 = mass / (height \* height) (mass in kg and height in meter).

**Your tasks:**

1. Store Mark's and John's mass and height in variables

2. Calculate both their BMIs using the formula (you can even implement both versions)

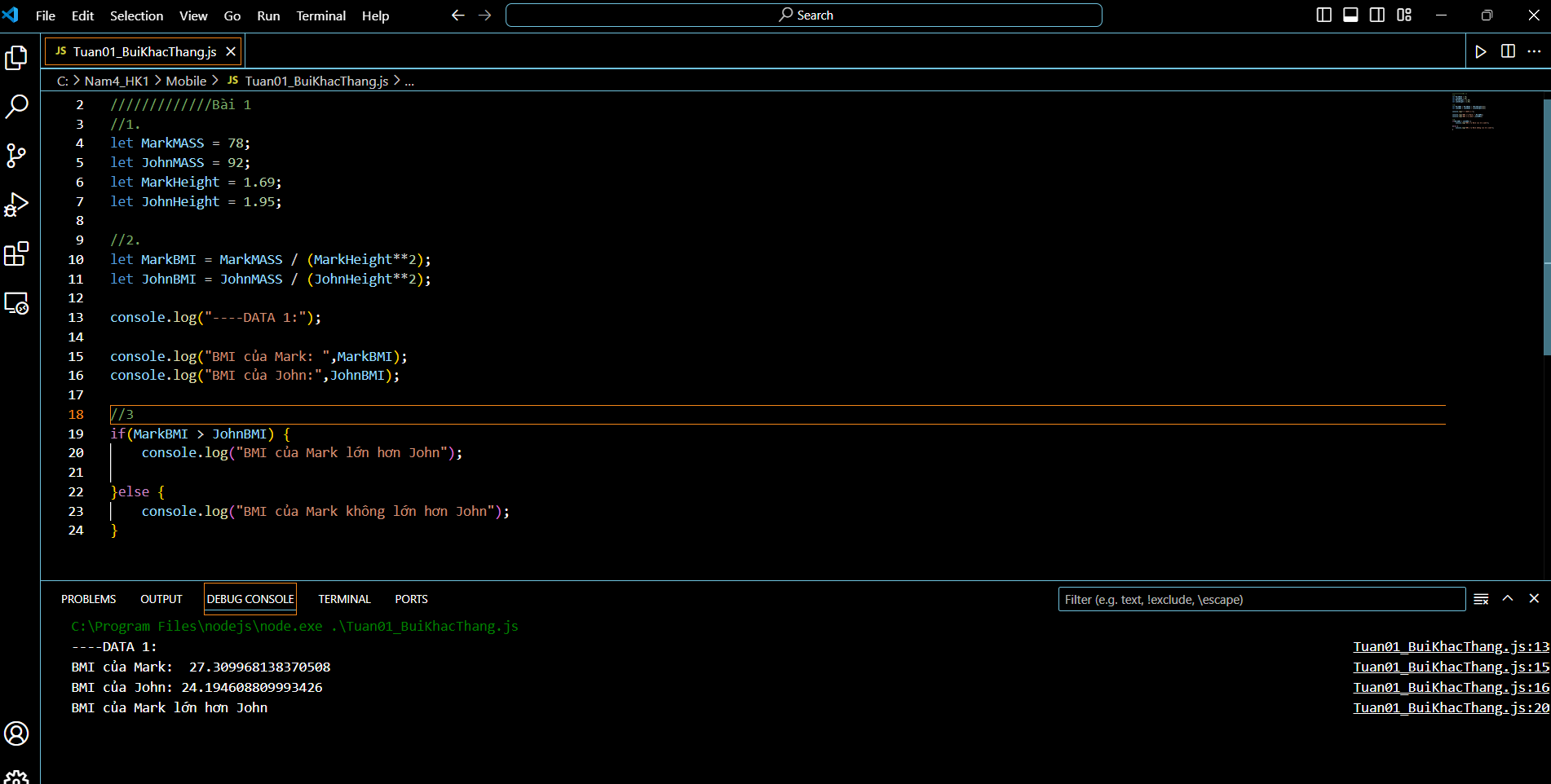
3. Create a Boolean variable 'markHigherBMI' containing information about

whether Mark has a higher BMI than John.

**Test data:**

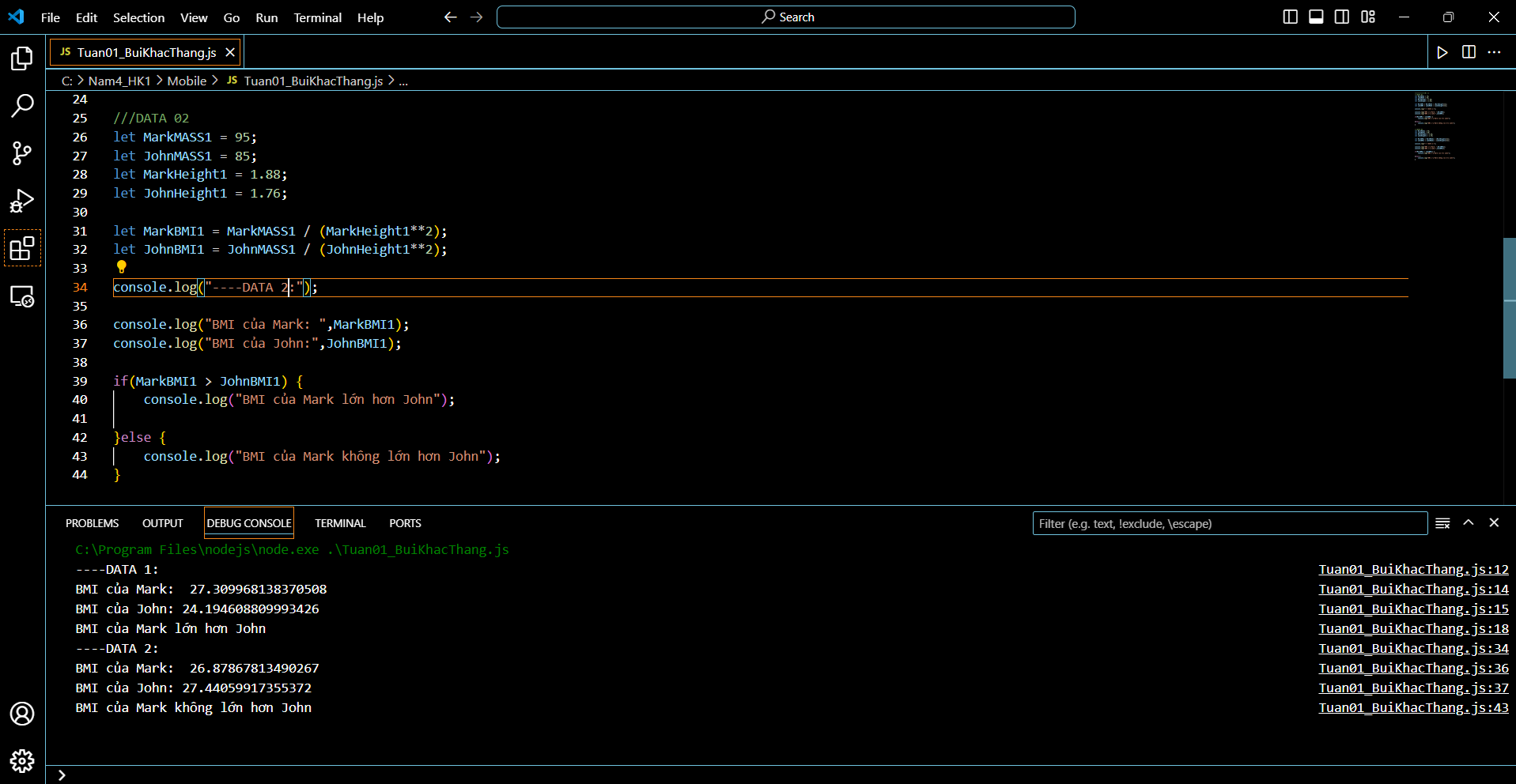
§ Data 1: Marks weights 78 kg and is 1.69 m tall. John weights 92 kg and is 1.95

m tall.



§ Data 2: Marks weights 95 kg and is 1.88 m tall. John weights 85 kg and is 1.76

m tall.



**Coding Challenge #2**

Use the BMI example from Challenge #1, and the code you already wrote, and improve it.

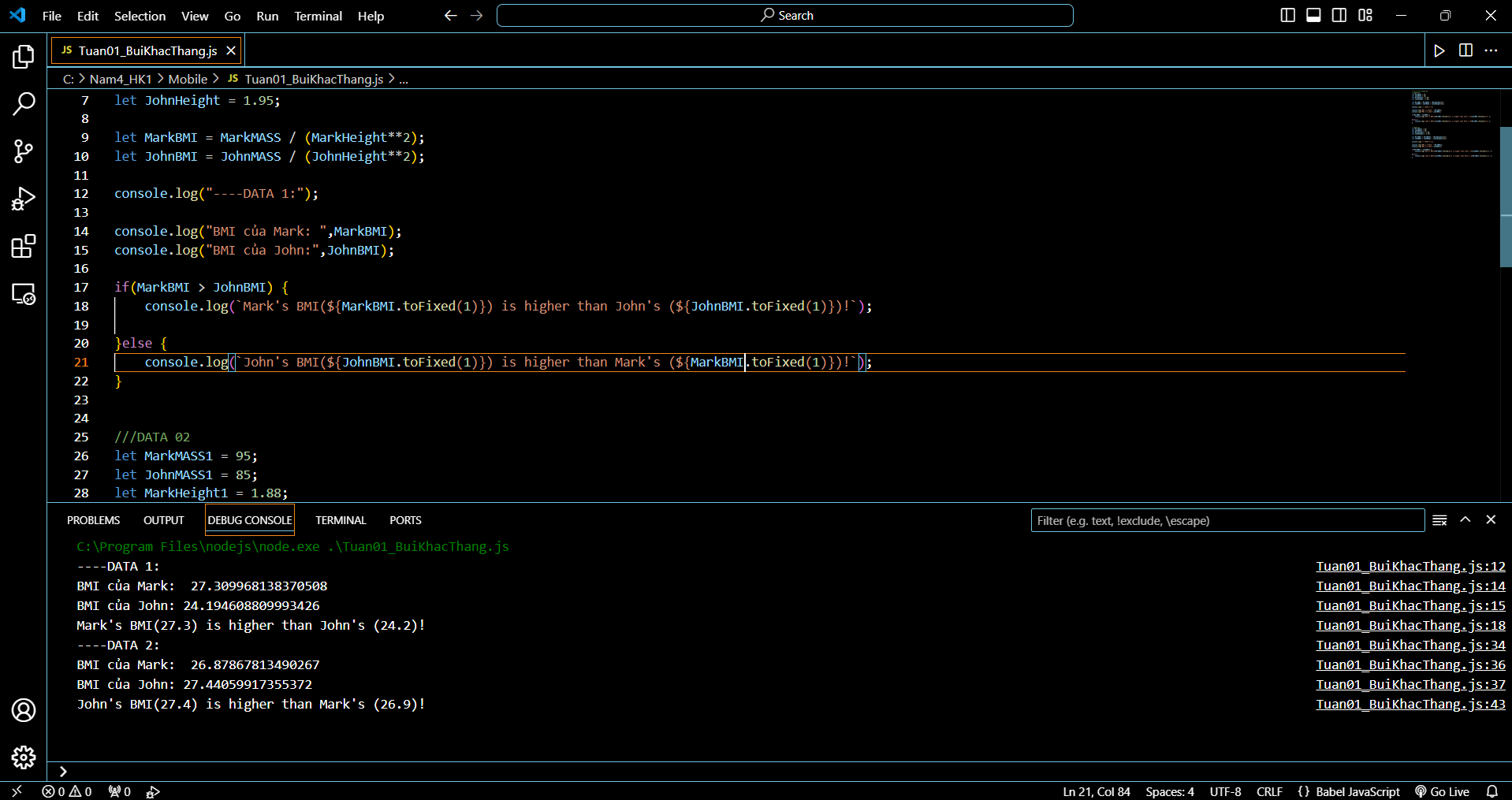
**Your tasks:**

1. Print a nice output to the console, saying who has the higher BMI. The message

is either "Mark's BMI is higher than John's!" or "John's BMI is higher than Mark's!"

2. Use a template literal to include the BMI values in the outputs. Example: "Mark's

BMI (28.3) is higher than John's (23.9)!"



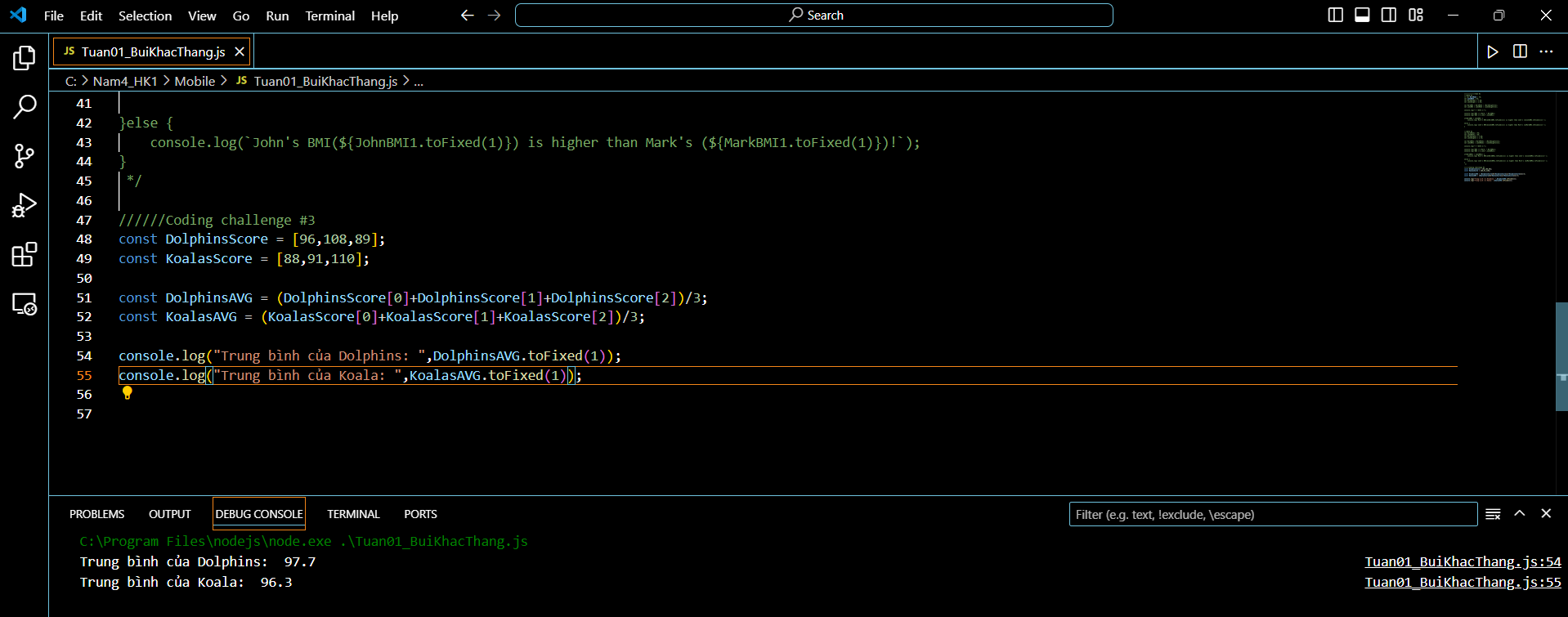
**Coding Challenge #3**

There are two gymnastics teams, Dolphins and Koalas. They compete against each

other 3 times. The winner with the highest average score wins a trophy!

**Your tasks:**

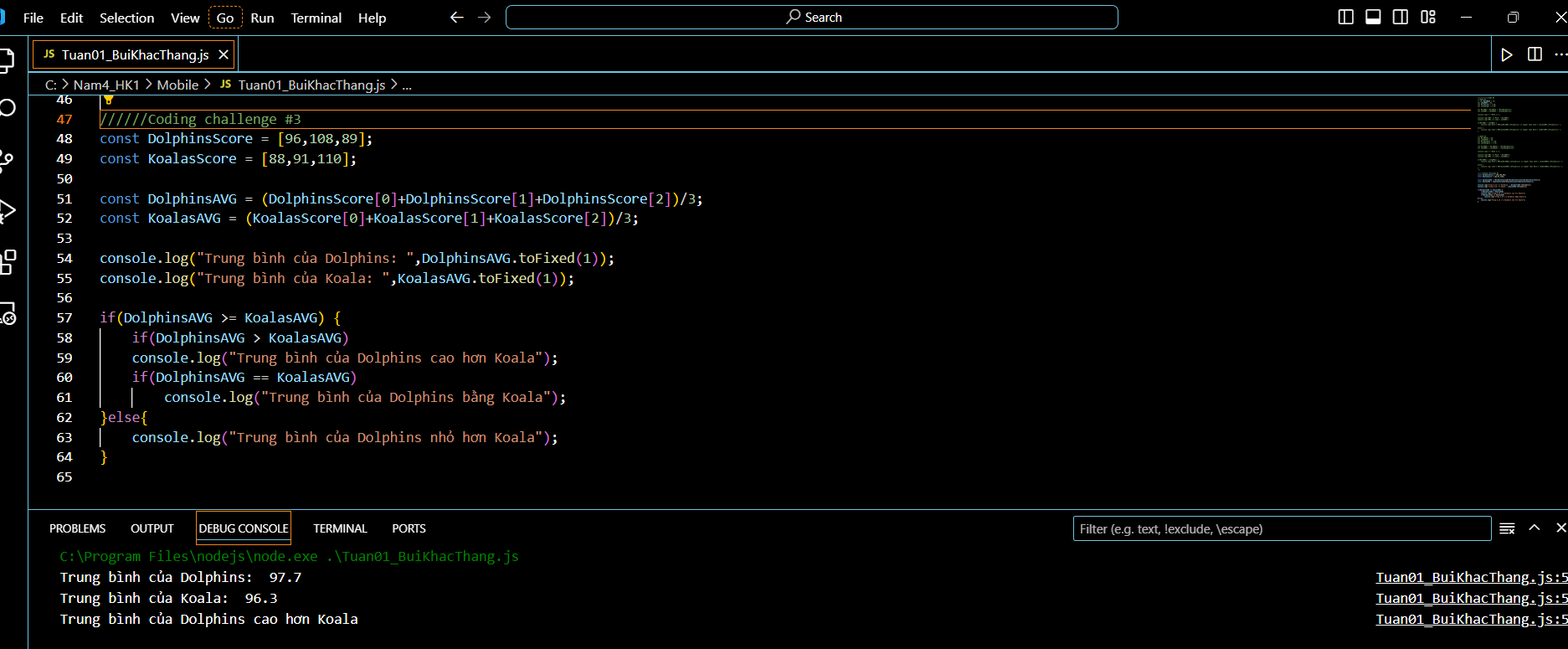
1. Calculate the average score for each team, using the test data below



2. Compare the team's average scores to determine the winner of the competition,

and print it to the console. Don't forget that there can be a draw, so test for that

as well (draw means they have the same average score)

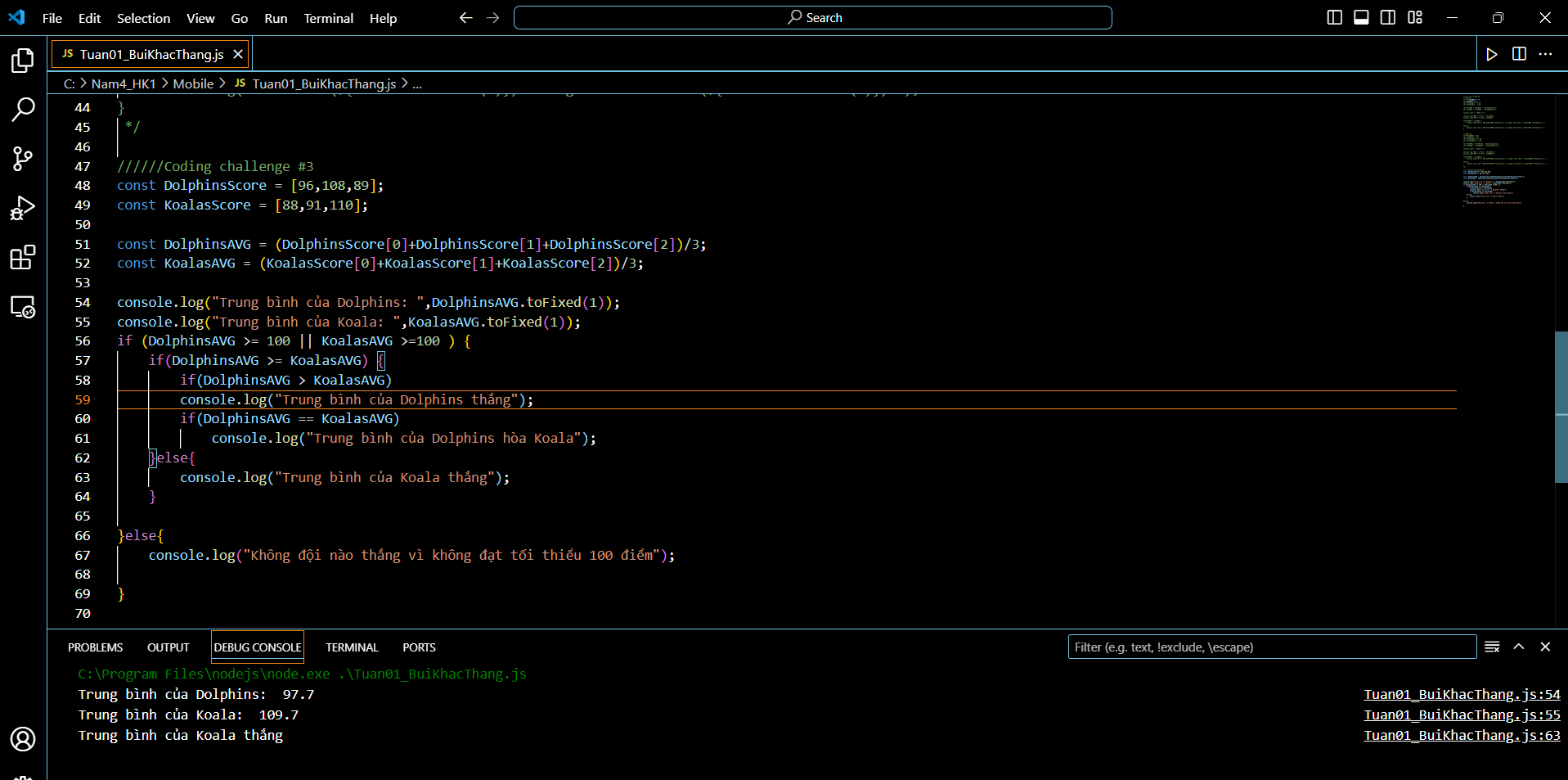


3. Bonus 1: Include a requirement for a minimum score of 100. With this rule, a

team only wins if it has a higher score than the other team, and the same time a

score of at least 100 points. Hint: Use a logical operator to test for minimum

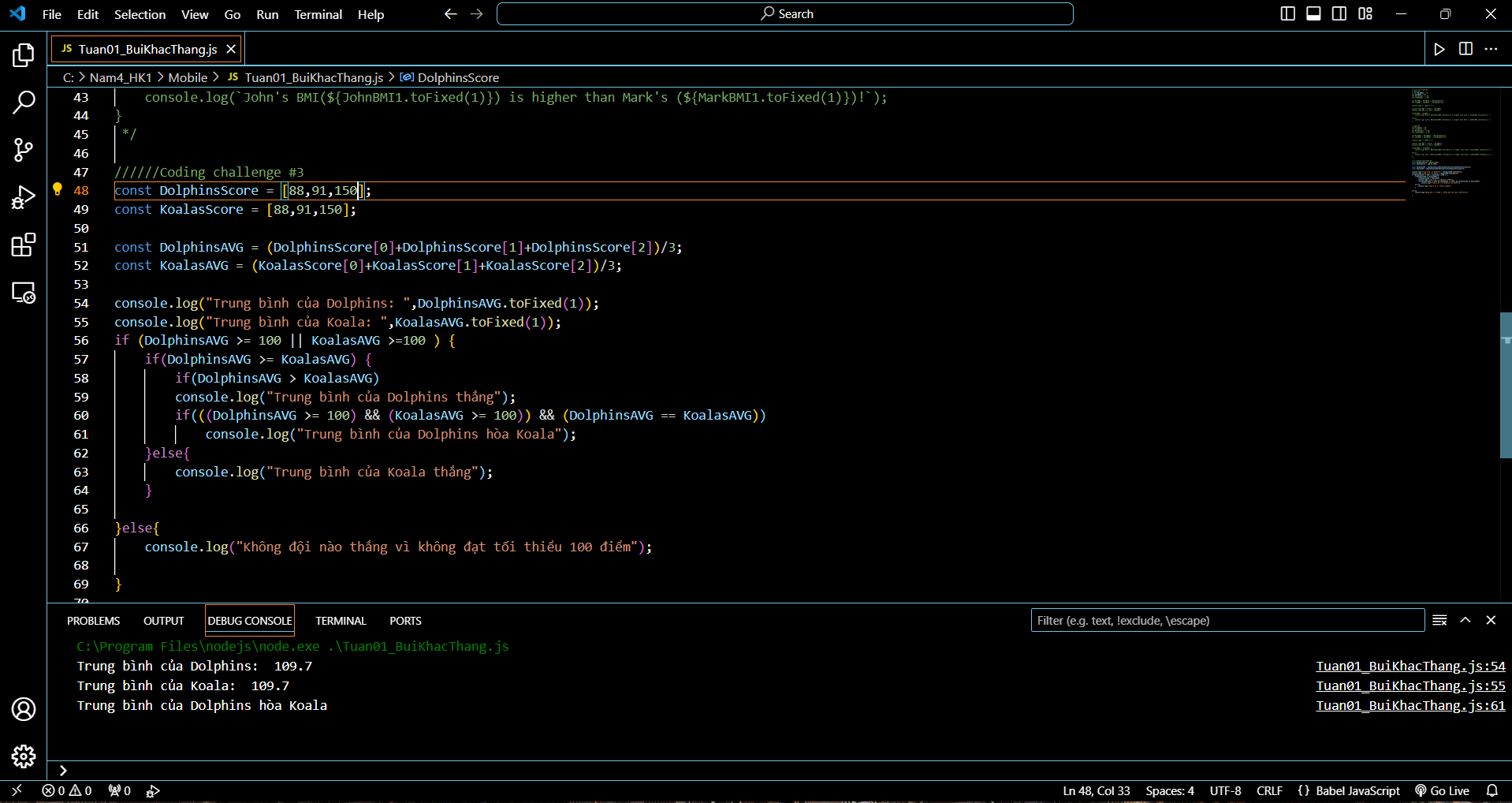
score, as well as multiple else-if blocks



4. Bonus 2: Minimum score also applies to a draw! So a draw only happens when

both teams have the same score and both have a score greater or equal 100

points. Otherwise, no team wins the trophy



**Test data:**

§ Data 1: Dolphins score 96, 108 and 89. Koalas score 88, 91 and 110

§ Data Bonus 1: Dolphins score 97, 112 and 101. Koalas score 109, 95 and 123

§ Data Bonus 2: Dolphins score 97, 112 and 101. Koalas score 109, 95 and 106

**Coding Challenge #4**

Steven wants to build a very simple tip calculator for whenever he goes eating in a

restaurant. In his country, it's usual to tip 15% if the bill value is between 50 and

300. If the value is different, the tip is 20%.

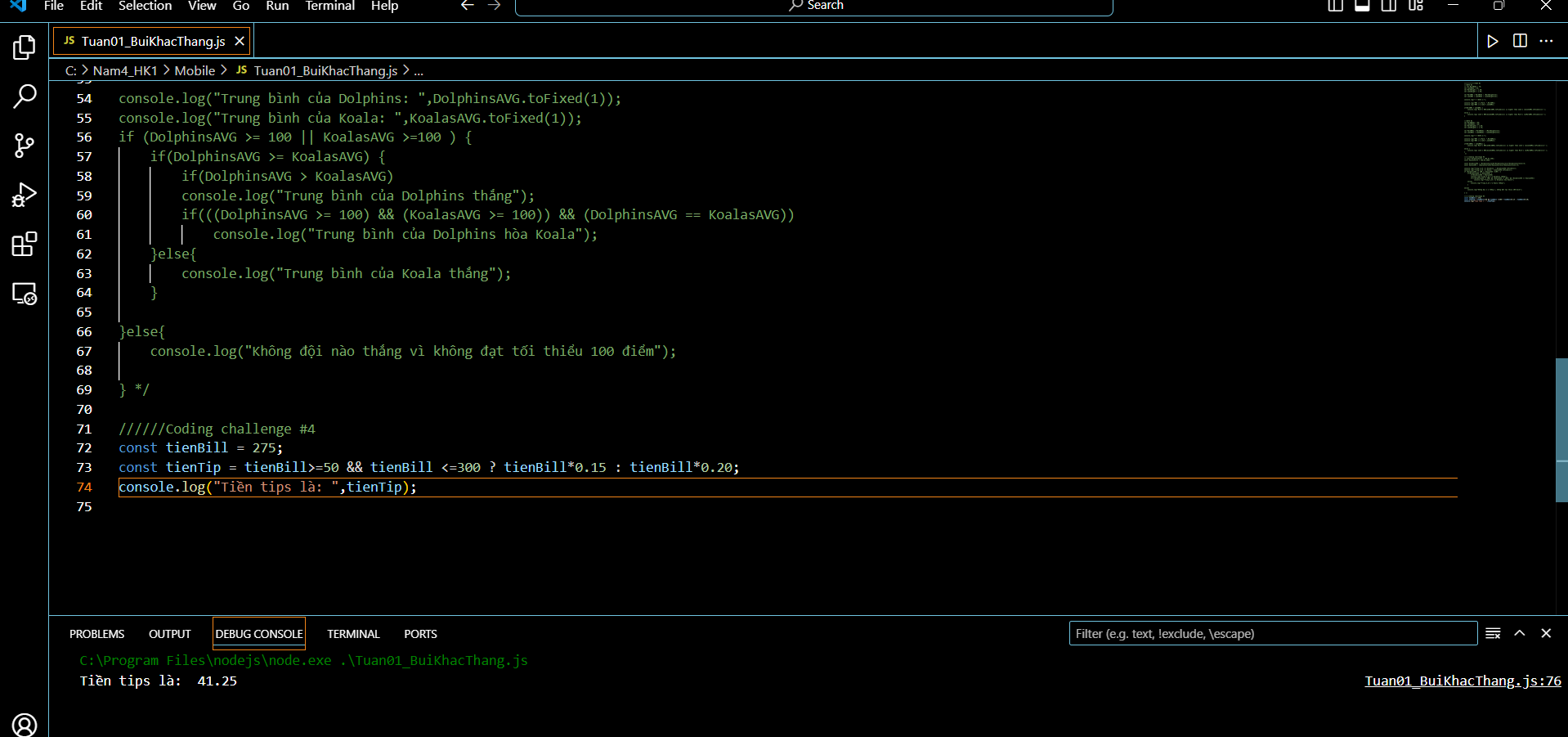
**Your tasks:**

1. Calculate the tip, depending on the bill value. Create a variable called 'tip' for

this. It's not allowed to use an if/else statement (If it's easier for you, you can

start with an if/else statement, and then try to convert it to a ternary

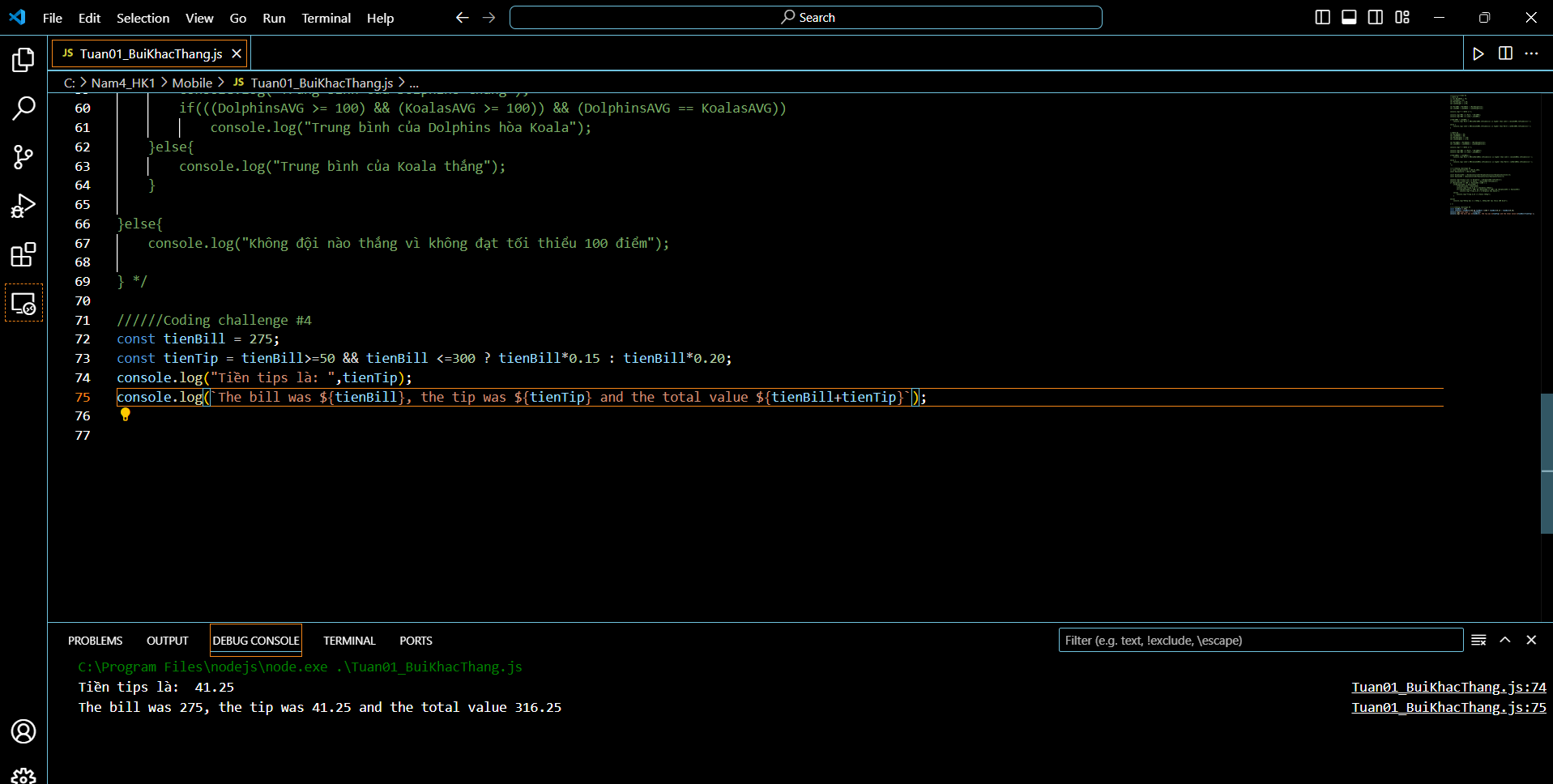
operator!)



2. Print a string to the console containing the bill value, the tip, and the final value

(bill + tip). Example: “The bill was 275, the tip was 41.25, and the total value

316.25”



**Test data:**

§ Data 1: Test for bill values 275, 40 and 430

Hints:

§ To calculate 20% of a value, simply multiply it by 20/100 = 0.2

§ Value X is between 50 and 300, if it's >= 50 && <= 300

**Coding Challenge #5**

Back to the two gymnastics teams, the Dolphins and the Koalas! There is a new

gymnastics discipline, which works differently.

Each team competes 3 times, and then the average of the 3 scores is calculated (so

one average score per team).

A team only wins if it has at least double the average score of the other team.

Otherwise, no team wins!

**Your tasks:**

1. Create an arrow function 'calcAverage' to calculate the average of 3 scores



1. Use the function to calculate the average for both teams

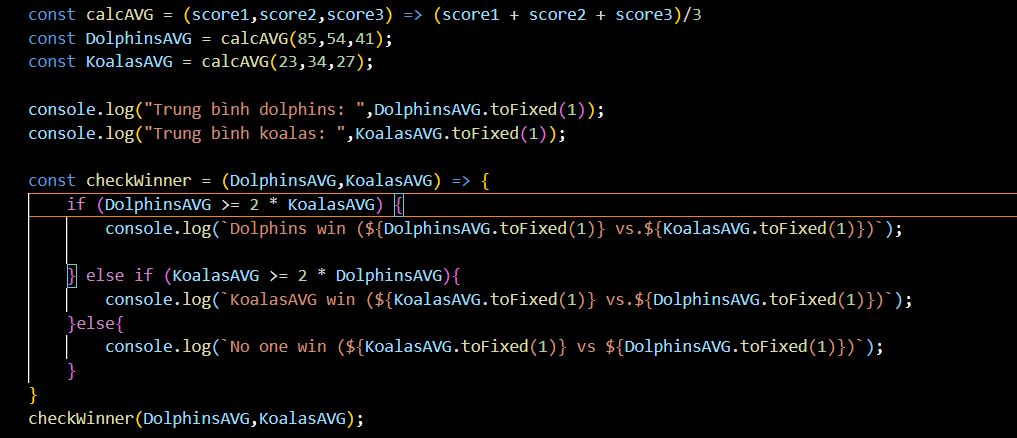


3. Create a function 'checkWinner' that takes the average score of each team

as parameters ('avgDolhins' and 'avgKoalas'), and then logs the winner

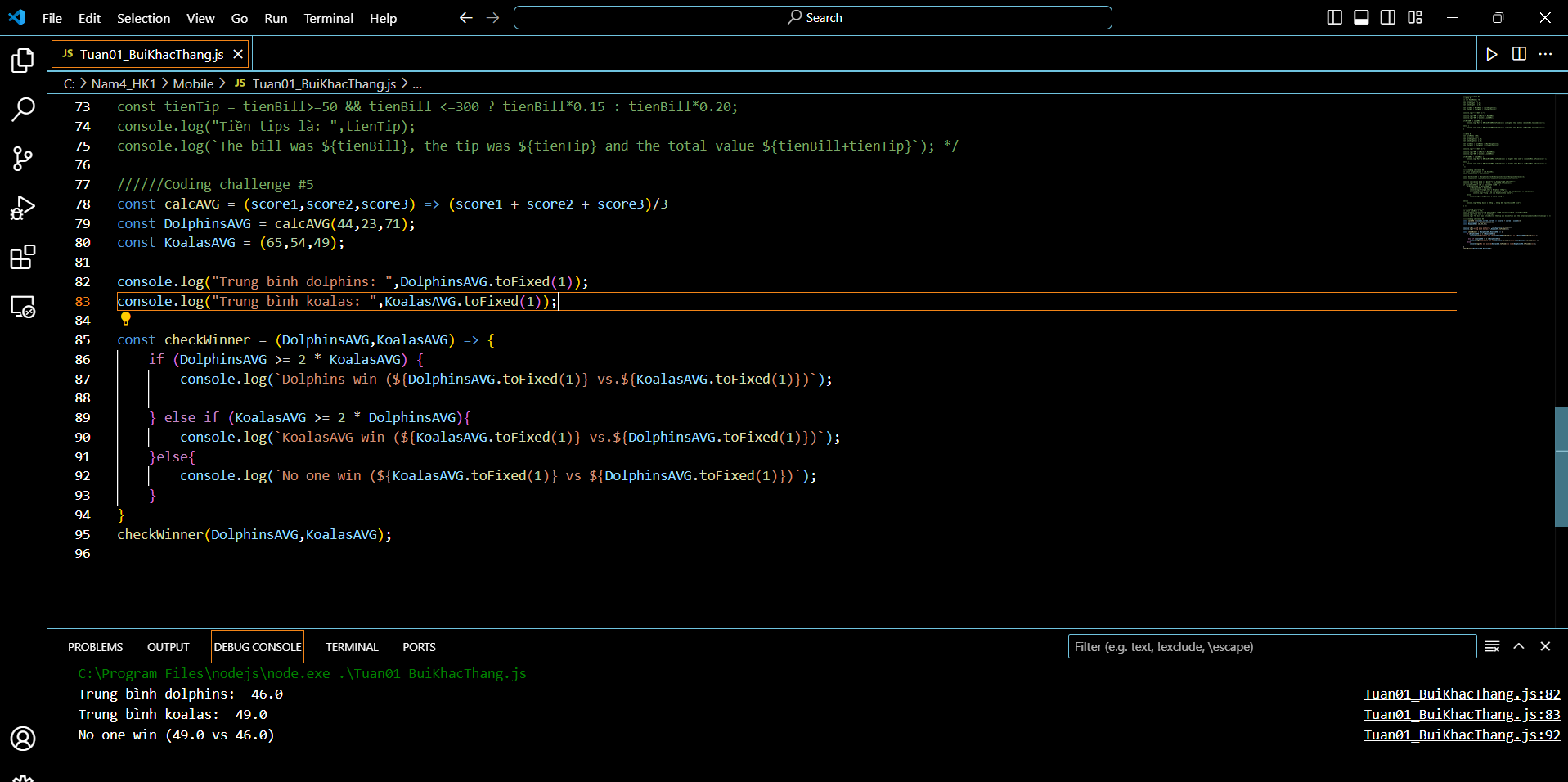
to the console, together with the victory points, according to the rule above.

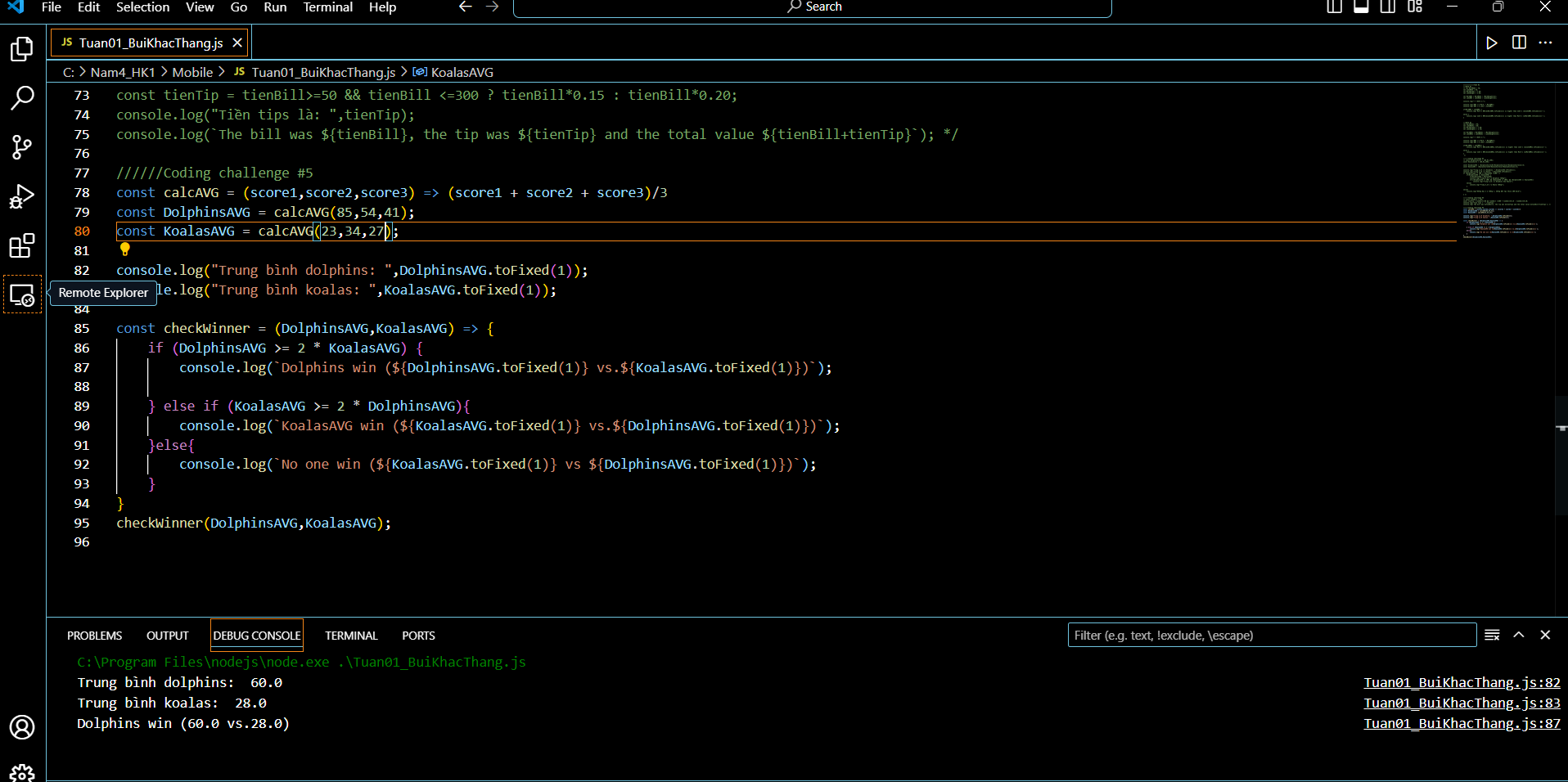
Example: "Koalas win (30 vs. 13)"



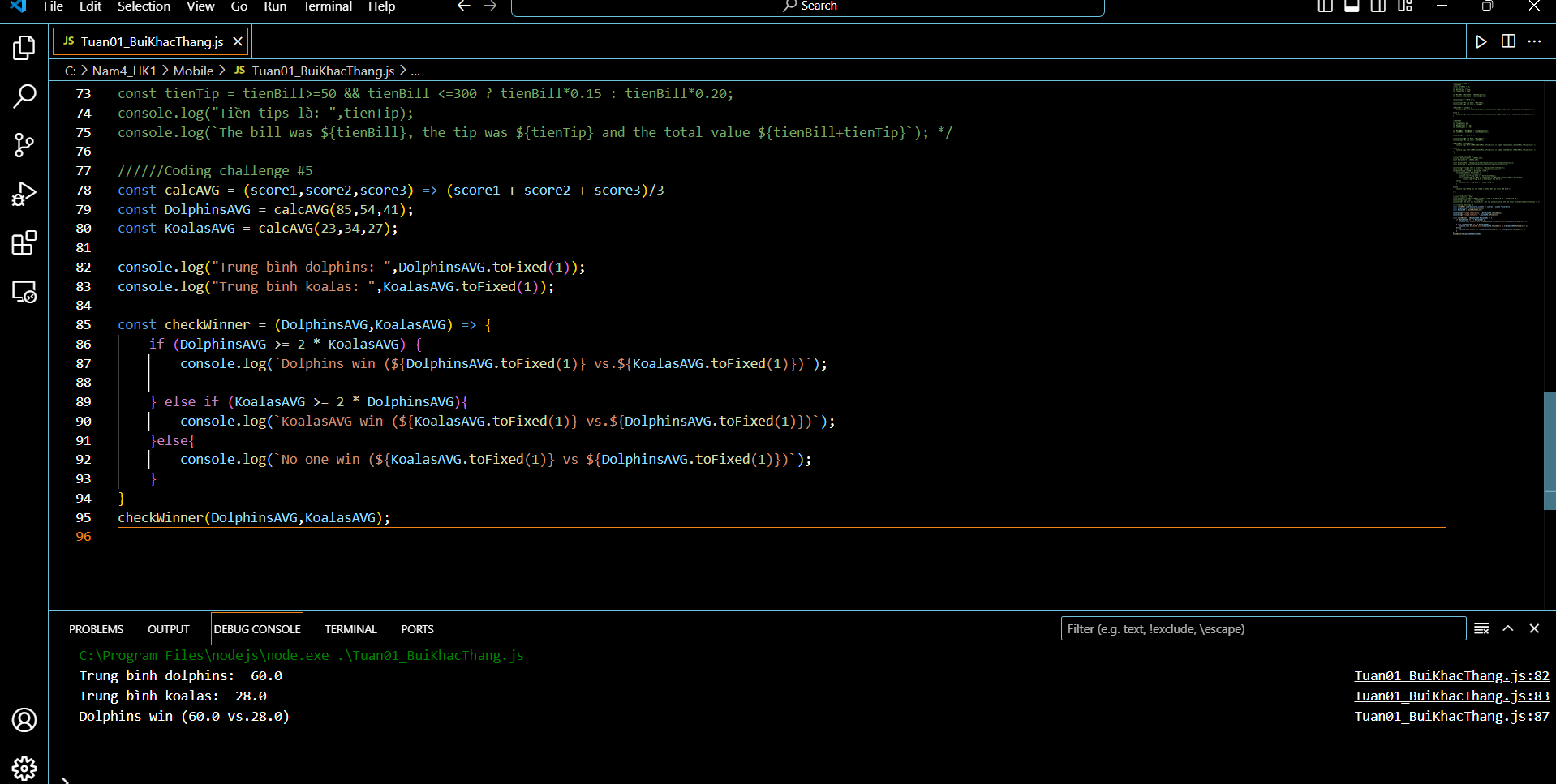
4. Use the 'checkWinner' function to determine the winner for both Data 1 and

Data 2





1. Ignore draws this time



**Test data:**

§ Data 1: Dolphins score 44, 23 and 71. Koalas score 65, 54 and 49

§ Data 2: Dolphins score 85, 54 and 41. Koalas score 23, 34 and 27

**Coding Challenge #6**

Steven is still building his tip calculator, using the same rules as before: Tip 15% of

the bill if the bill value is between 50 and 300, and if the value is different, the tip is

20%.

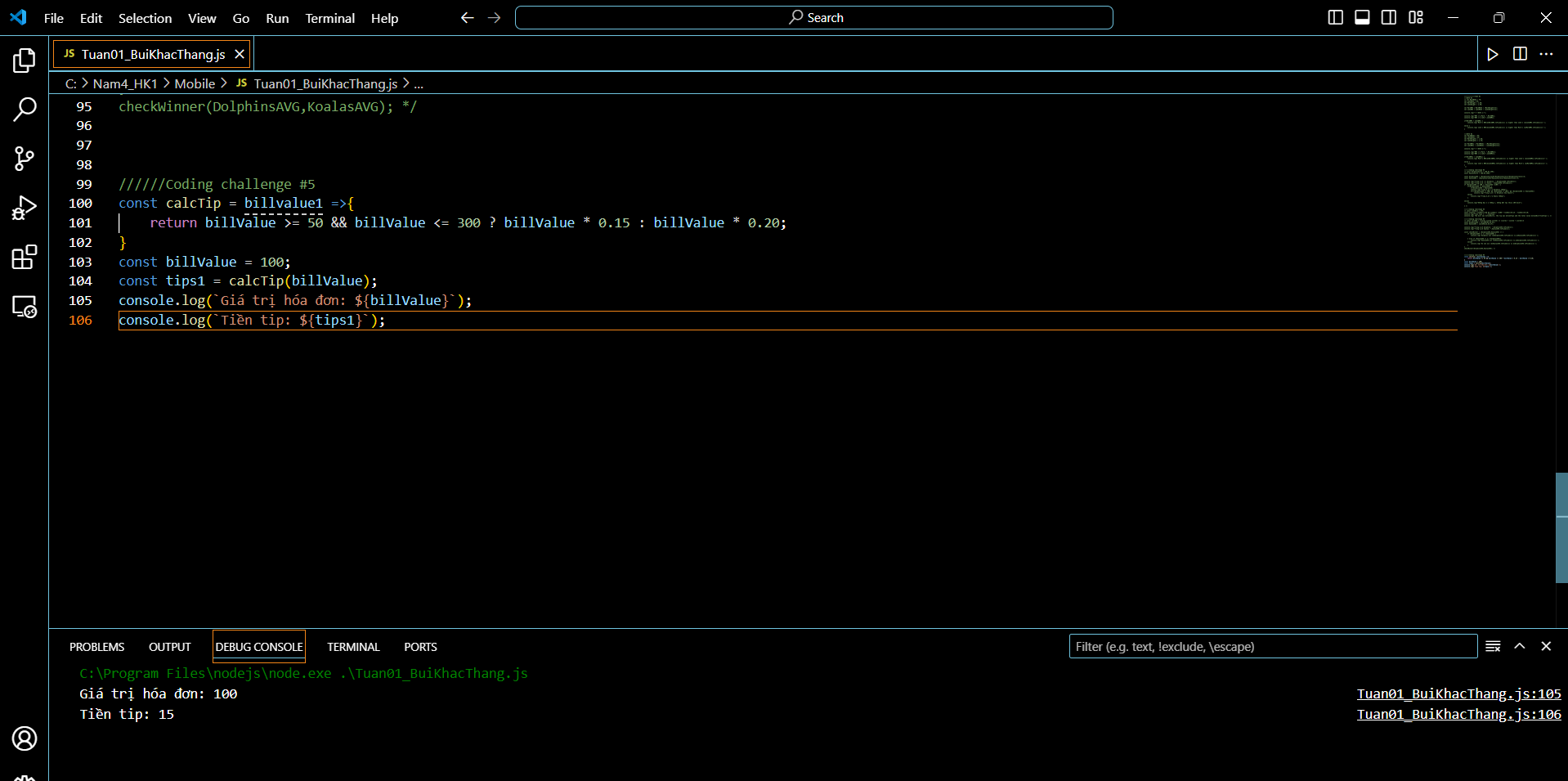
**Your tasks:**

1. Write a function 'calcTip' that takes any bill value as an input and returns

the corresponding tip, calculated based on the rules above (you can check out

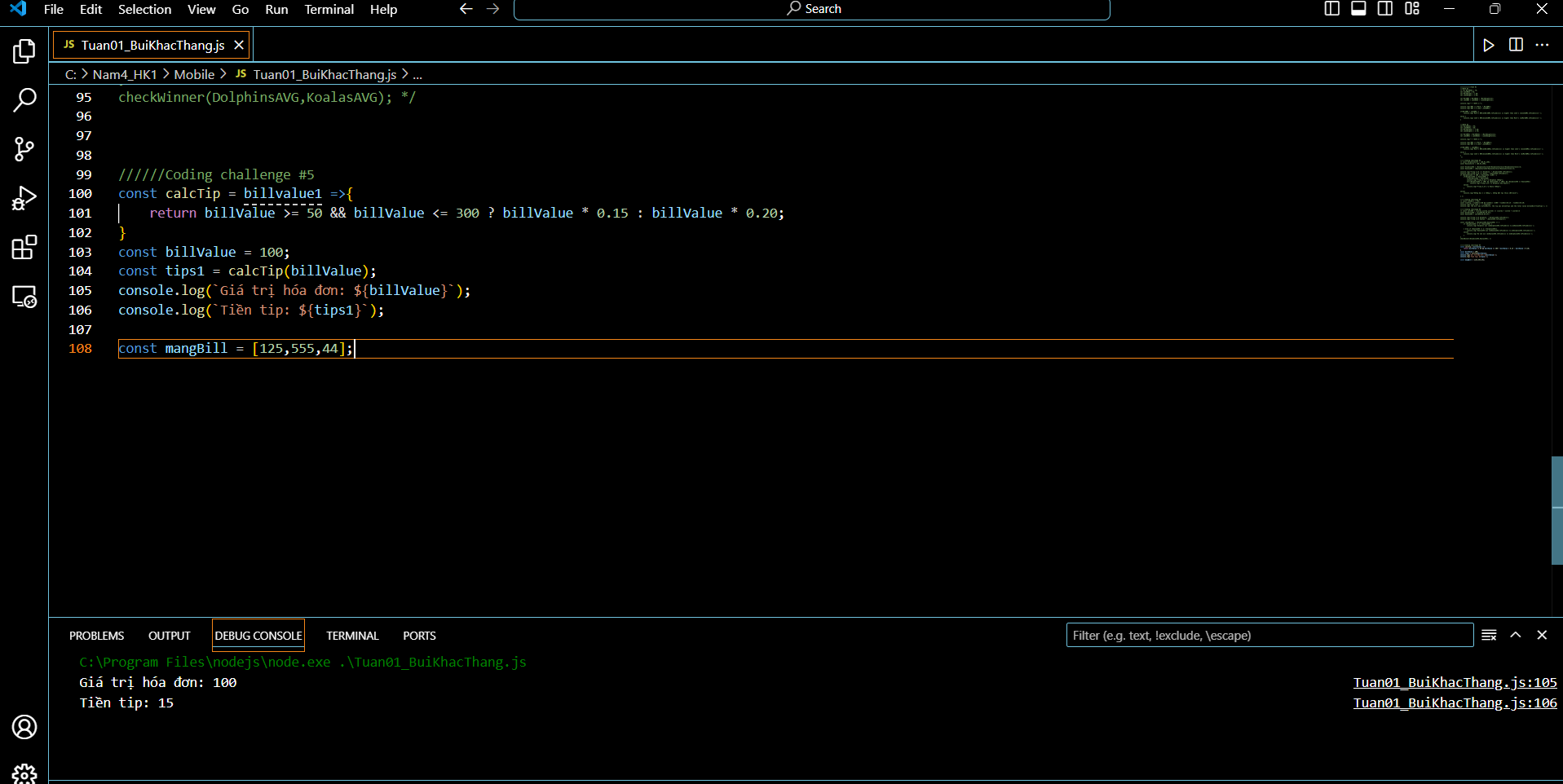
the code from first tip calculator challenge if you need to). Use the function

type you like the most. Test the function using a bill value of 100



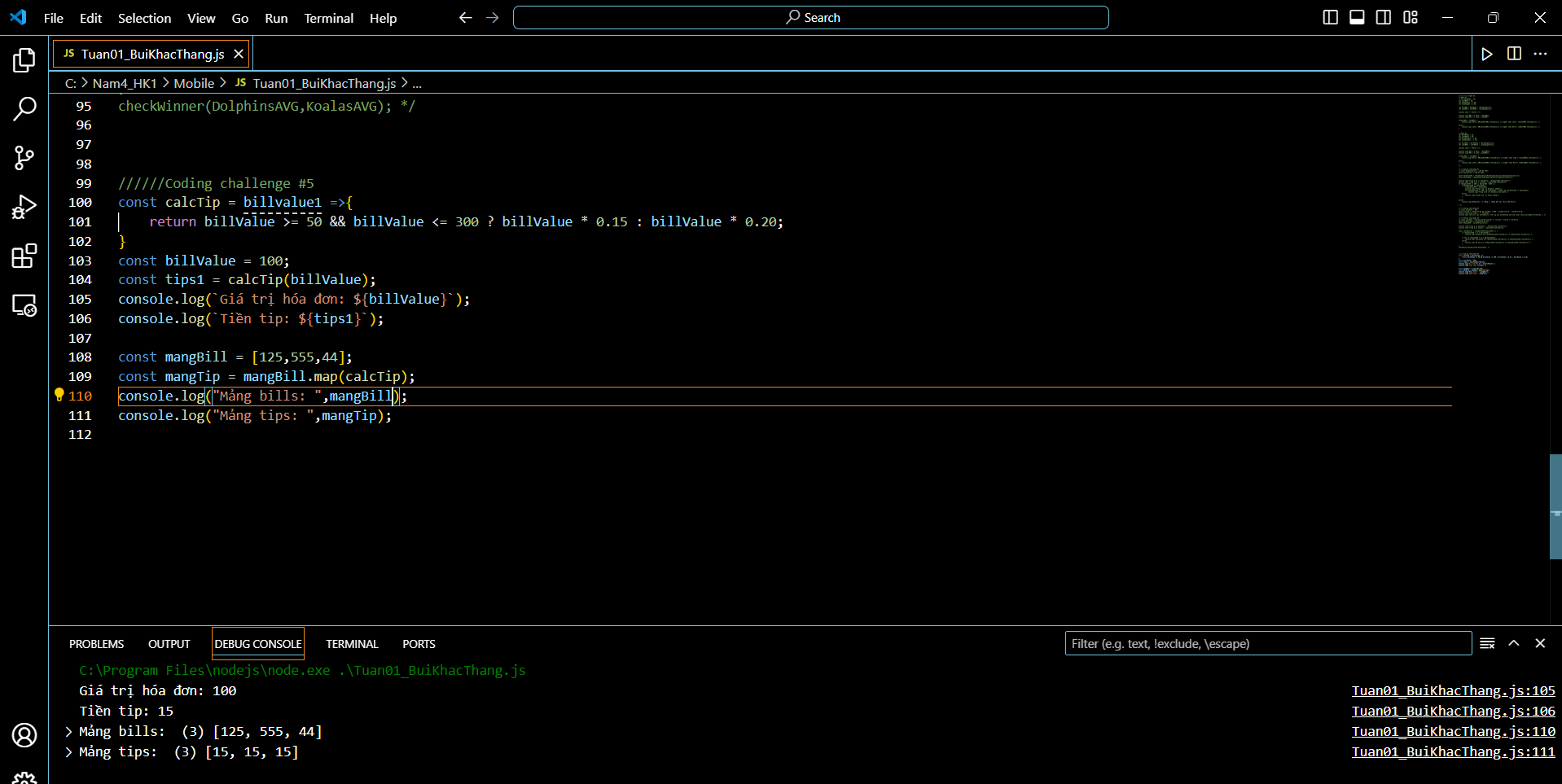
2. And now let's use arrays! So create an array 'bills' containing the test data

Below



3. Create an array 'tips' containing the tip value for each bill, calculated from

the function you created before



1. Bonus: Create an array 'total' containing the total values, so the bill + tip



**Test data:** 125, 555 and 44

**Coding Challenge #7**

Let's go back to Mark and John comparing their BMIs! This time, let's use objects to

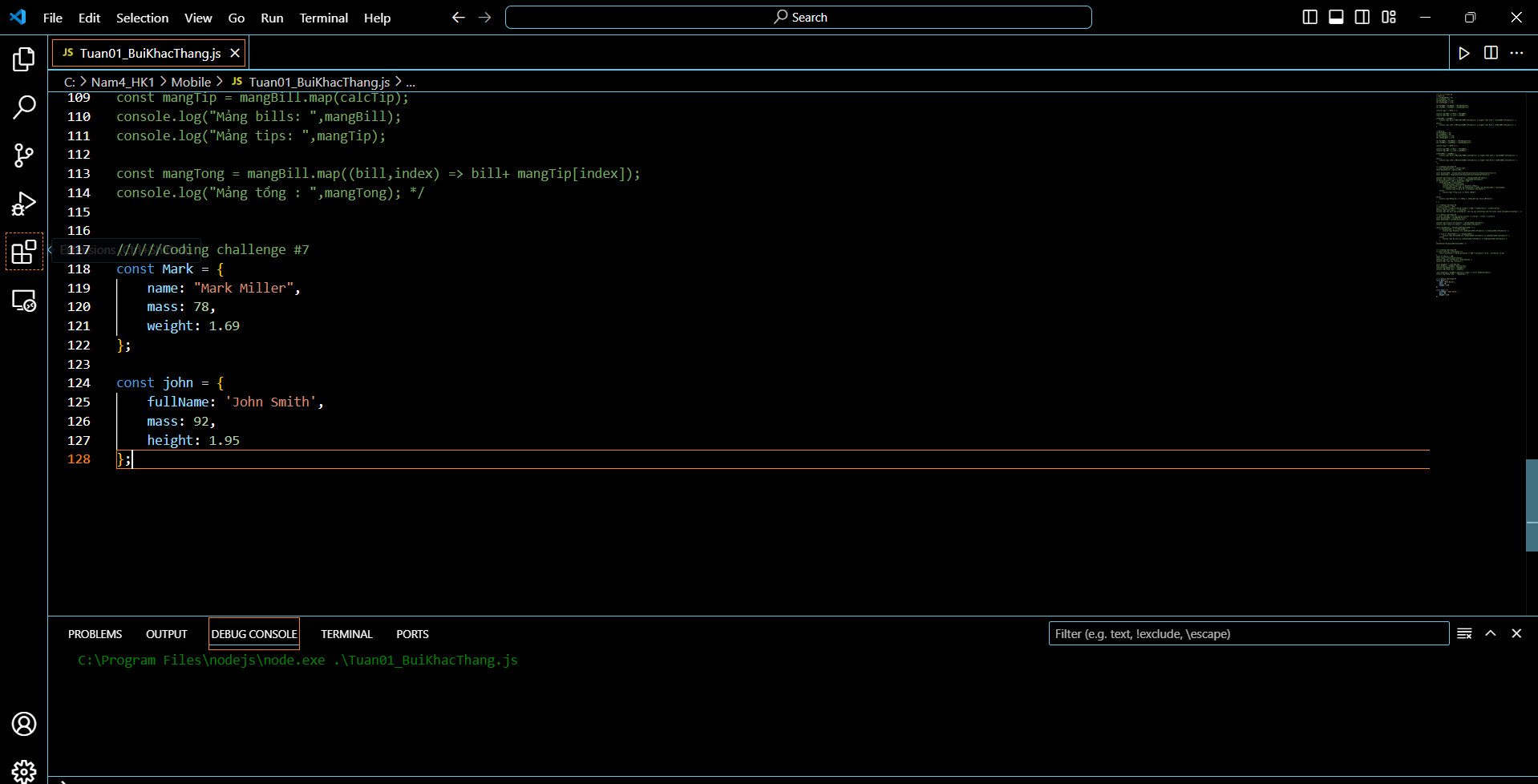
implement the calculations! Remember: BMI = mass / height \*\* 2 = mass

/ (height \* height) (mass in kg and height in meter)

**Your tasks:**

1. For each of them, create an object with properties for their full name, mass, and

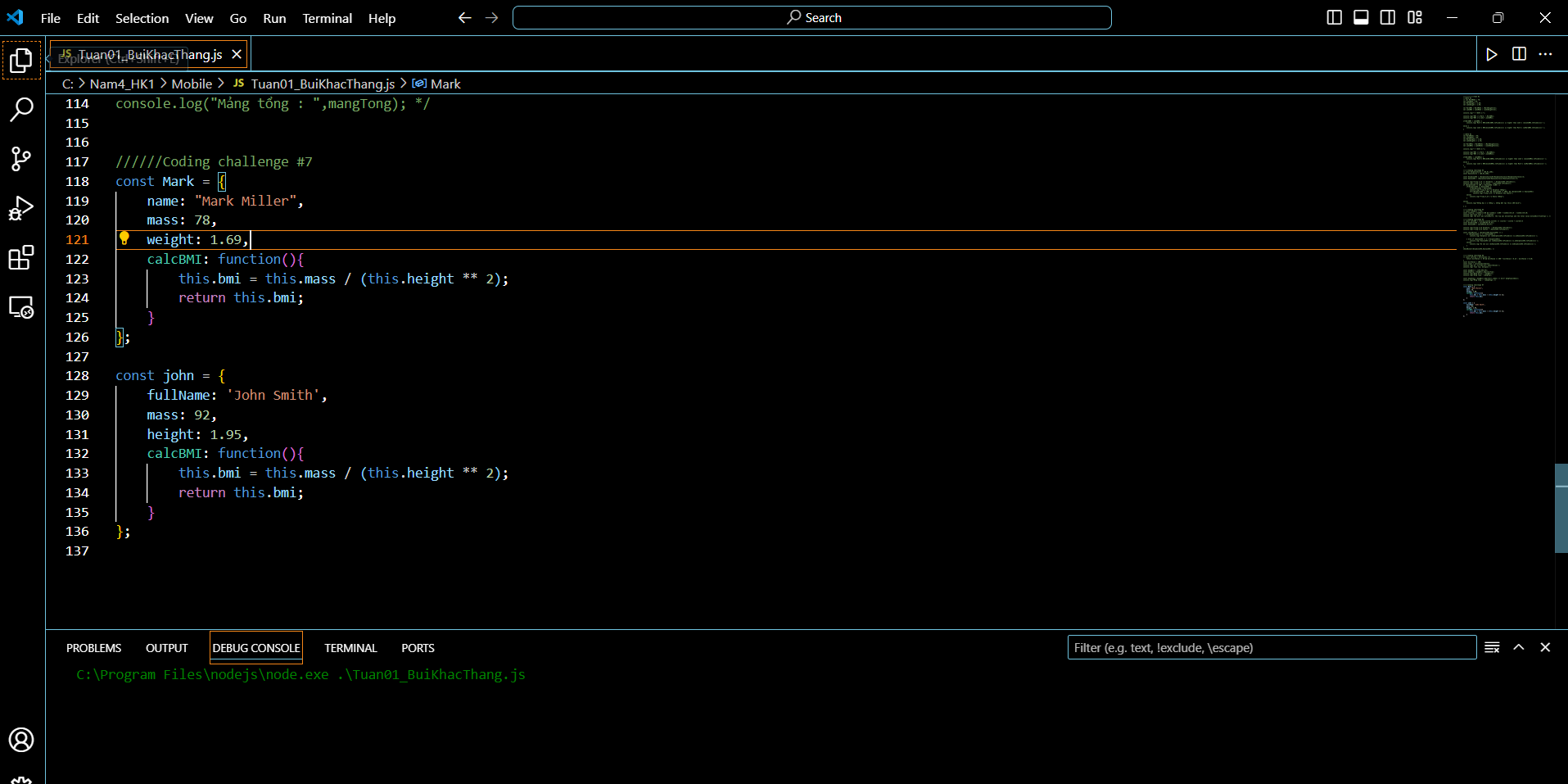
height (Mark Miller and John Smith)



2. Create a 'calcBMI' method on each object to calculate the BMI (the same

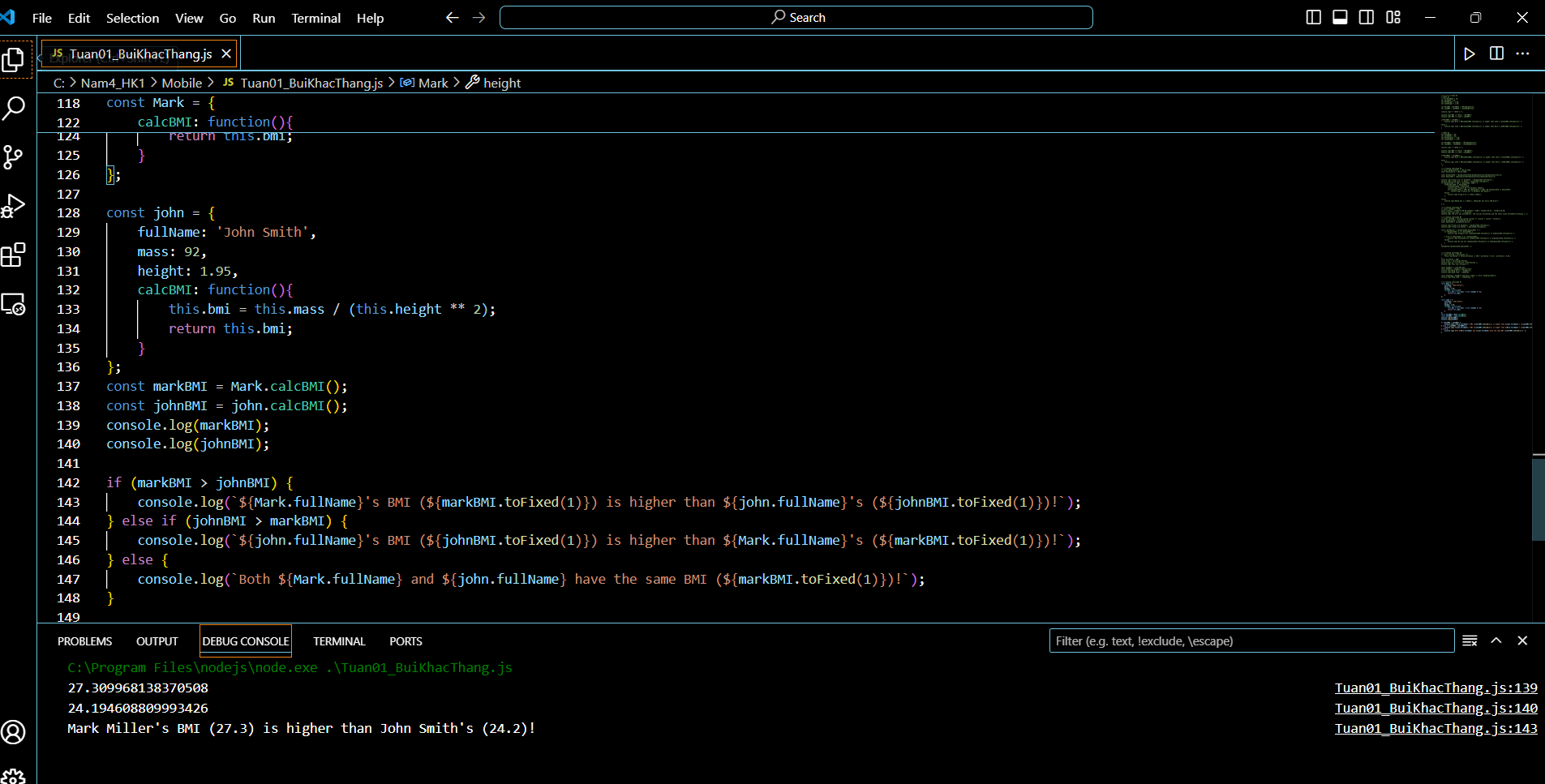
method on both objects). Store the BMI value to a property, and also return it

from the method



3. Log to the console who has the higher BMI, together with the full name and the

respective BMI. Example: "John's BMI (28.3) is higher than Mark's (23.9)!"



**Test data:** Marks weights 78 kg and is 1.69 m tall. John weights 92 kg and is 1.95 m

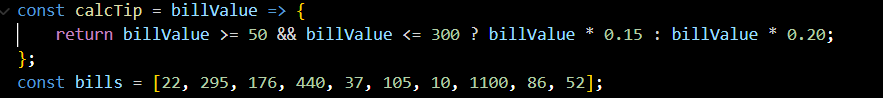
tall.

**Coding Challenge #8**

Let's improve Steven's tip calculator even more, this time using loops!

Your tasks:

1. Create an array 'bills' containing all 10 test bill values



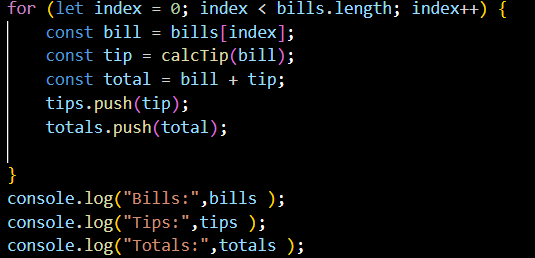
1. Create empty arrays for the tips and the totals ('tips' and 'totals')



3. Use the 'calcTip' function we wrote before (no need to repeat) to calculate

tips and total values (bill + tip) for every bill value in the bills array. Use a for

loop to perform the 10 calculations!



Test data: 22, 295, 176, 440, 37, 105, 10, 1100, 86 and 52

Hints: Call ‘calcTip ‘in the loop and use the push method to add values to the

tips and totals arrays



**Bonus:**

4. Bonus: Write a function 'calcAverage' which takes an array called 'arr' as

an argument. This function calculates the average of all numbers in the given

array. This is a difficult challenge (we haven't done this before)! Here is how to

solve it:

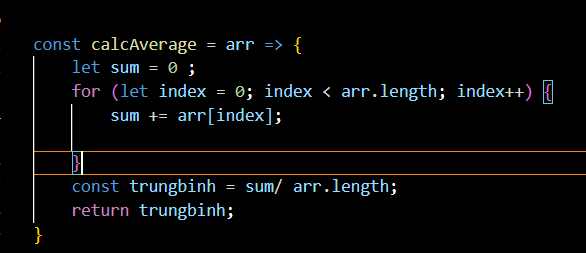
4.1. First, you will need to add up all values in the array. To do the addition,

start by creating a variable 'sum' that starts at 0. Then loop over the

array using a for loop. In each iteration, add the current value to the

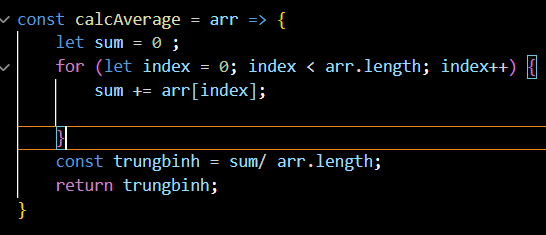
'sum' variable. This way, by the end of the loop, you have all values

added together

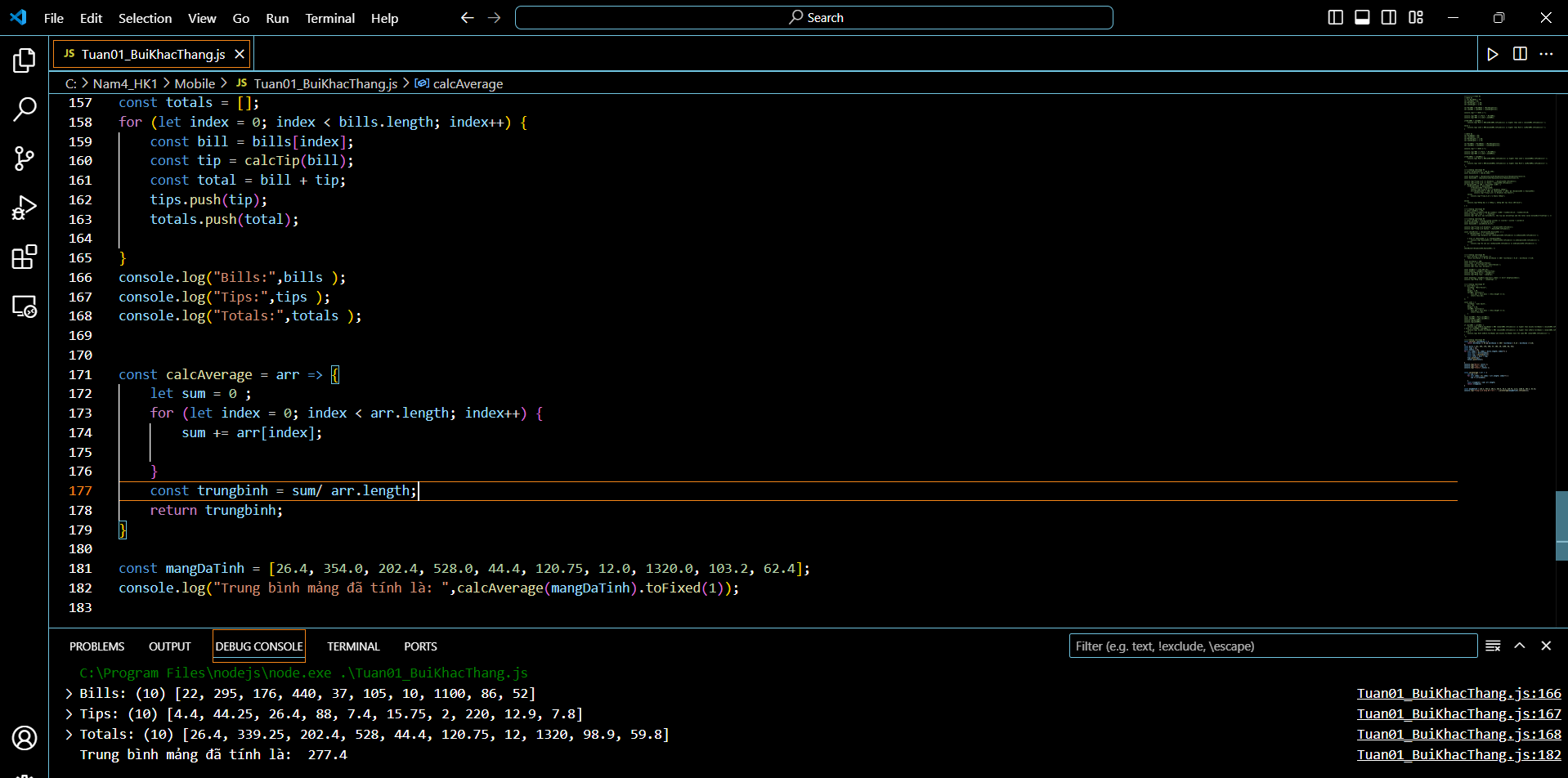


4.2. To calculate the average, divide the sum you calculated before by the

length of the array (because that's the number of elements)



4.3. Call the function with the 'totals' array



**Coding Challenge #9**

Given an array of forecasted maximum temperatures, the thermometer displays a

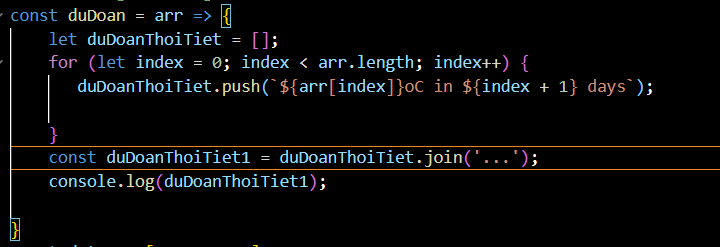
string with the given temperatures. Example: [17, 21, 23] will print "... 17oC in 1

days ... 21oC in 2 days ... 23oC in 3 days ..."

**Your tasks:**

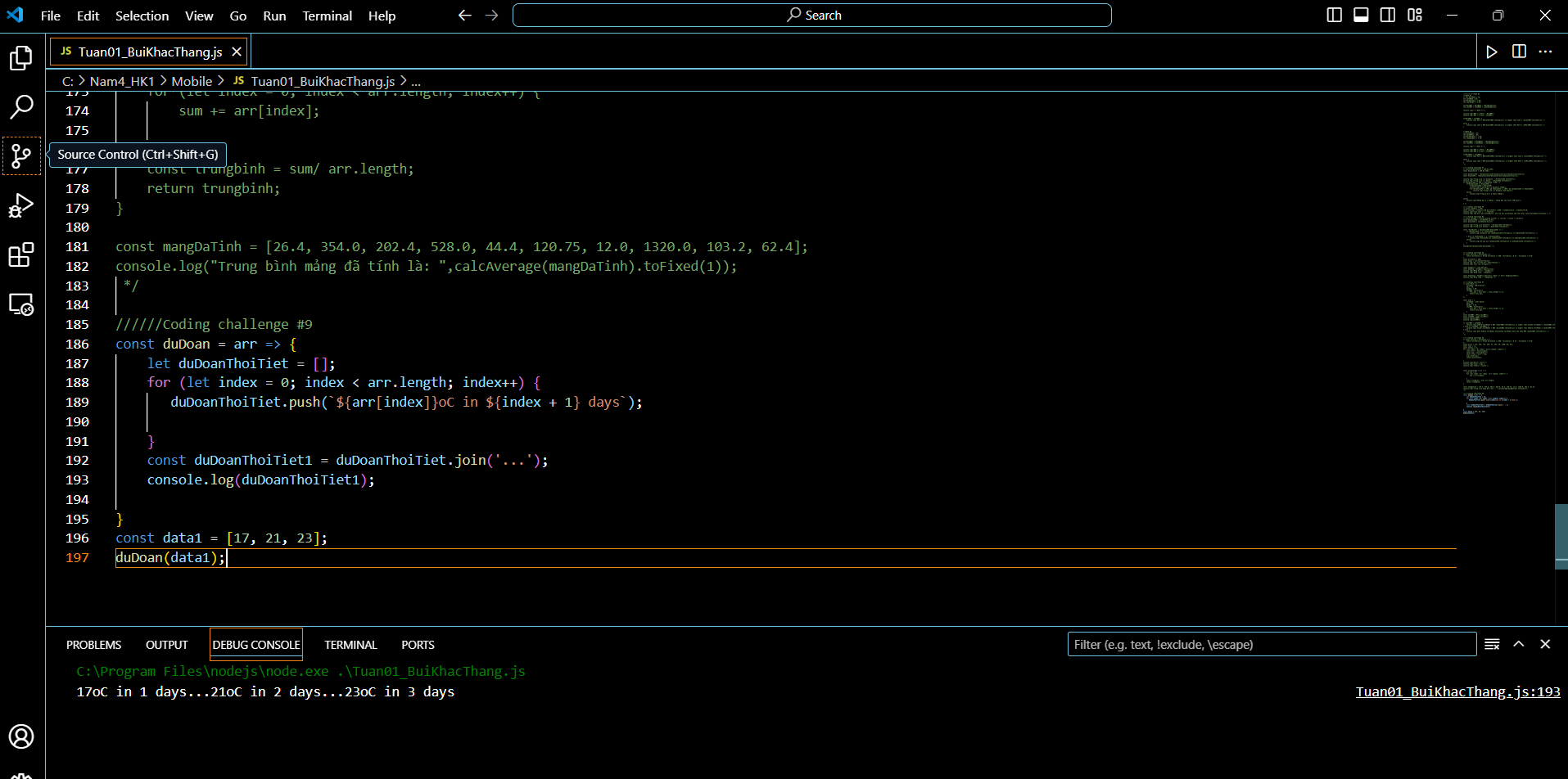
1. Create a function 'printForecast' which takes in an array 'arr' and logs a

string like the above to the console. Try it with both test datasets.



2. Use the problem-solving framework: Understand the problem and break it up

into sub-problems!



**Test data:**

§ Data 1: [17, 21, 23]

§ Data 2: [12, 5, -5, 0, 4]



**JavaScript in the Browser: DOM and Events**

**Coding Challenge #1**

Implement a game rest functionality, so that the player can make a new guess!

**Your tasks:**

1. Select the element with the 'again' class and attach a click event handler

2. In the handler function, restore initial values of the 'score' and

'secretNumber' variables

3. Restore the initial conditions of the message, number, score and guess input

fields

4. Also restore the original background color (#222) and number width (15rem)

**Data Structures, Modern Operators and Strings**

**Coding Challenge #1**

We're building a football betting app (soccer for my American friends )! Suppose we get data from a web service about a certain game ('game' variable on next page). In this challenge we're gonna work with that data.

**Your tasks:**

1. Create one player array for each team (variables 'players1' and 'players2')

2. The first player in any player array is the goalkeeper and the others are field players. For Bayern Munich (team 1) create one variable ('gk') with the goalkeeper's name, and one array ('fieldPlayers') with all the remaining 10 field players

3. Create an array 'allPlayers' containing all players of both teams (22 players)

4. During the game, Bayern Munich (team 1) used 3 substitute players. So create a new array ('players1Final') containing all the original team1 players plus 'Thiago', 'Coutinho' and 'Perisic'

5. Based on the game.odds object, create one variable for each odd (called 'team1', 'draw' and 'team2')

6. Write a function ('printGoals') that receives an arbitrary number of player names (not an array) and prints each of them to the console, along with the number of goals that were scored in total (number of player names passed in)

7. The team with the lower odd is more likely to win. Print to the console which team is more likely to win, without using an if/else statement or the ternary operator.

**Test data for 6**: First, use players 'Davies', 'Muller', 'Lewandowski' and 'Kimmich'.

Then, call the function again with players from game.scored

**Coding Challenge #2**

Let's continue with our football betting app! Keep using the 'game' variable from

before.

**Your tasks:**

1. Loop over the game.scored array and print each player name to the console,

along with the goal number (Example: "Goal 1: Lewandowski")

2. Use a loop to calculate the average odd and log it to the console (We already

studied how to calculate averages, you can go check if you don't remember)

3. Print the 3 odds to the console, but in a nice formatted way, exactly like this:

Odd of victory Bayern Munich: 1.33

Odd of draw: 3.25

Odd of victory Borrussia Dortmund: 6.5

Get the team names directly from the game object, don't hardcode them

(except for "draw"). Hint: Note how the odds and the game objects have the

same property names

4. Bonus: Create an object called 'scorers' which contains the names of the

players who scored as properties, and the number of goals as the value. In this

game, it will look like this:

{

Gnarby: 1,

Hummels: 1,

Lewandowski: 2

}

**Coding Challenge #3**

Let's continue with our football betting app! This time, we have a map called

'gameEvents' (see below) with a log of the events that happened during the

game. The values are the events themselves, and the keys are the minutes in which

each event happened (a football game has 90 minutes plus some extra time).

**Your tasks:**

1. Create an array 'events' of the different game events that happened (no duplicates)

2. After the game has finished, is was found that the yellow card from minute 64 was unfair. So remove this event from the game events log.

3. Compute and log the following string to the console: "An event happened, on average, every 9 minutes" (keep in mind that a game has 90 minutes)

4. Loop over 'gameEvents' and log each element to the console, marking whether it's in the first half or second half (after 45 min) of the game, like this:

[FIRST HALF] 17: ⚽ GOAL

**Coding Challenge #4**

Write a program that receives a list of variable names written in underscore\_case

and convert them to camelCase.

The input will come from a textarea inserted into the DOM (see code below to

insert the elements), and conversion will happen when the button is pressed.

**Test data (pasted to textarea, including spaces):**

underscore\_case

first\_name

Some\_Variable

calculate\_AGE

delayed\_departure

Should produce this output (5 separate console.log outputs):

underscoreCase ✅

firstName ✅✅

someVariable ✅✅✅

calculateAge ✅✅✅✅

delayedDeparture ✅✅✅✅✅

Hints:

§ Remember which character defines a new line in the textarea

§ The solution only needs to work for a variable made out of 2 words, like a\_b

§ Start without worrying about the ✅. Tackle that only after you have the variable

name conversion working

§ This challenge is difficult on purpose, so start watching the solution in case

you're stuck. Then pause and continue!

Afterwards, test with your own test data!