```
/*******************************
* CODING CHALLENGE 2
*/

/*
John and Mike both play basketball in different
teams. In the latest 3 games, John's team scored 89,
120 and 103 points, while Mike's team scored 116, 94
and 123 points.

1. Calculate the average score for each team
2. Decide which teams wins in average (highest
average score), and print the winner to the console.
Also include the average score in the output.
3. Then change the scores to show different winners.
Don't forget to take into account there might be a
draw (the same average score)

4. EXTRA: Mary also plays basketball, and her team
scored 97, 134 and 105 points. Like before, log the
average winner to the console. HINT: you will need
the && operator to take the decision. If you can't
solve this one, just watch the solution, it's no
problem :)
5. Like before, change the scores to generate
different winners, keeping in mind there might be
draws.

GOOD LUCK 😊
*/
```

```
// JavaScript problem solve

//1
var johnTeam = (140 + 120 + 103) / 3;
var mikeTeam = (119 + 94 + 123) / 3;
console.log(johnTeam, mikeTeam);


//2 & 3
if( johnTeam > mikeTeam) {
    console.log('John team wins with ' + johnTeam + ' points');
}
else if (mikeTeam > johnTeam) {
    console.log('Mike team wins with ' + mikeTeam + ' points');
}
else {
    console.log('There scores are same');
}
```

```
//4
var johnTeam = (89 + 120 + 103) / 3;
var mikeTeam = (119 + 94 + 123) / 3;
var maryTeam = (97 + 134 + 105) / 3;
console.log(johnTeam, mikeTeam, maryTeam);

if (johnTeam > mikeTeam && johnTeam > maryTeam){
    console.log('john team wins ' + johnTeam + ' points');
}
else if (mikeTeam > johnTeam && mikeTeam > maryTeam){
    console.log('Mike team wins ' + mikeTeam + ' points');
}
else if (maryTeam > johnTeam && maryTeam > mikeTeam) {
    console.log('Mary team wins with ' + maryTeam + ' points')
}
else {
    console.log('There score is same');
}
```

```
const myCalculator = function (bill) {
    var percentages;

    if (bill < 50) {
        percentages = .2;
    }
    else if (bill >= 50 && bill < 200) {
        percentages = 15 / 100;
    }
    else {
        percentages = .1;
    }

    return percentages * bill;
}

var bills = [124, 48, 268];
var tips = [
    myCalculator(bills[0]),
    myCalculator(bills[1]),
    myCalculator(bills[2])];


var finalValues = [
    bills[0] + tips[0],
    bills[1] + tips[1],
    bills[2] + tips[2],

];
console.log(tips, finalValues);


var mila = {
    firstName: 'mila',
    LastName: 'mili',
    family: ['jon', 'mark']

}
```

```
console.log(mila.firstName);
console.log(mila['LastName']);
mila['family'] = true;
console.log(mila)


var john = {
    firstName: 'John',
    lastName: 'smit',
    birth: 1990,
    calAge: function(){
        this.age = 2012 - this.birth;
    }
};
john.calAge();
console.log(john);
console.log(john.calAge(1290));
```

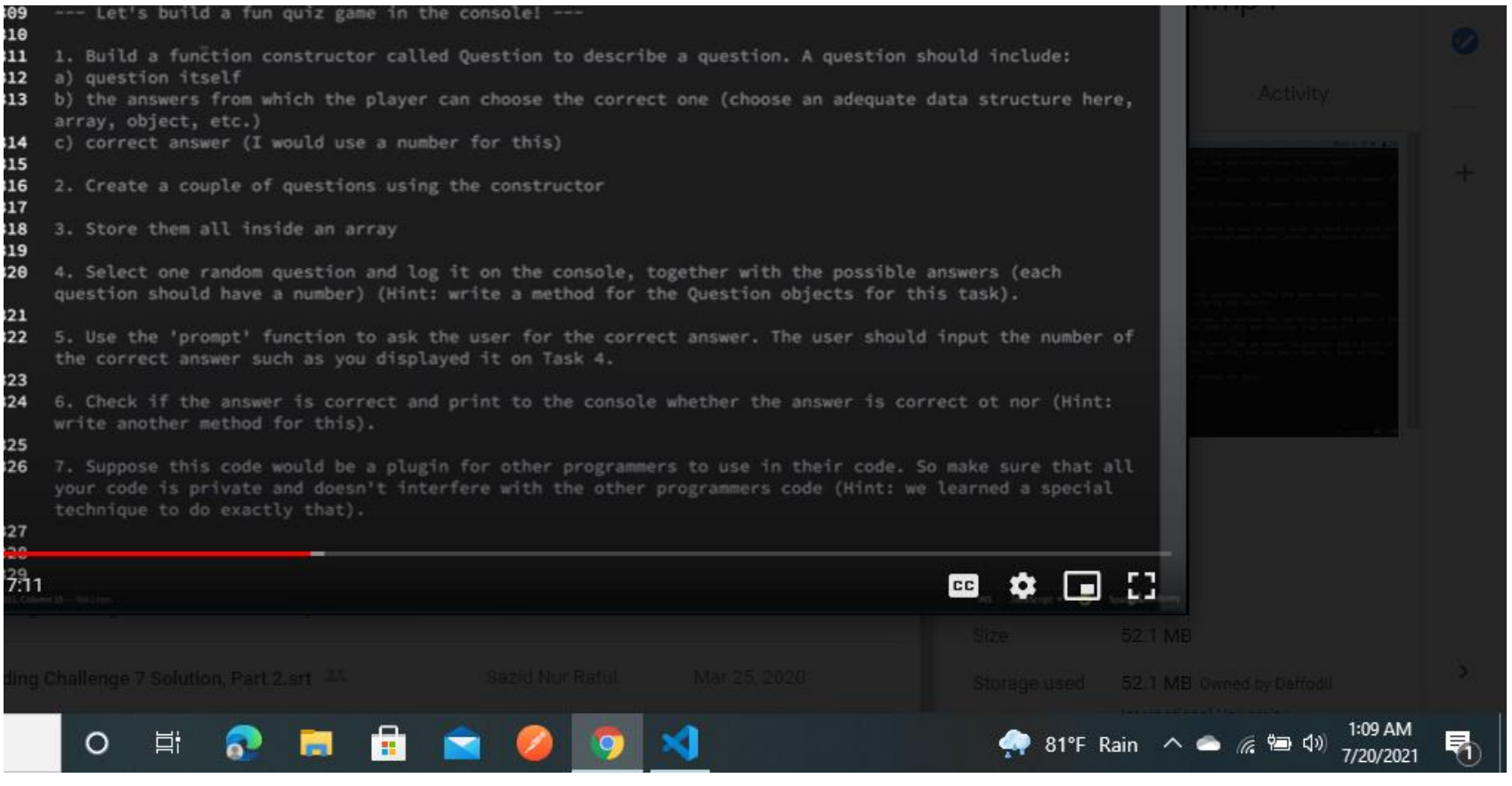
/Coding challenge 1

```
* 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).

1. Store Mark's and John's mass and height in
variables
2. Calculate both their BMIs
3. Create a boolean variable containing information
about whether Mark has a higher BMI than John.
4. Print a string to the console containing the
variable from step 3. (Something like "Is Mark's BMI
higher than John's? true").

GOOD LUCK 😊
*/
```

```
//code challange
const john = {
    johnfullName: 'John smit',
    mass: 123,
    height: 1.2,
    calBMI: function() {
        this.bmi = (this.mass / (this.height * this.height));
         return this.bmi;
    }
}
const mark = {
    markfullName: 'mark smit',
    mass: 13,
    height: 1.9,
    calBMI: function() {
        this.bmi = (this.mass / (this.height * this.height));
         return this.bmi;
    }
}
// mark.calBMI();
// john.calBMI();
// console.log(john, mark);
```

```
if ( john.calBMI() > mark.calBMI()){
    console.log(john.johnfullName + ' bmi high' + john.bmi);

}else if(mark.bmi > mark.bmi){
    console.log(mark.markfullName + ' bmi high' + mark.bmi);

}else{
    console.log('both are same');
}


var john = ['sfgg','dfg','dfsgg'];
for(var i = 0; i<3; i++ ){
    console.log(john);
}


var john = {
    firstName: 'john',
   bills: [124, 48, 268, 180],
   calTips: function(){
       this.tips = [];
       this.finlValues = [];


       for (var i =0; i < this.bills.length; i++);
       {
        var percentage;
        var bill =this.bills[i];
        if(bill < 50){
            percentage = 20/100;
        }else if (bill >= 50 && bill < 200){
            percentage = 15/100;
        }else{
            percentage = 10/100;
        }
        this.tips[i] = bill * percentage;
        this.finlValues[i] = bill + bill * percentage;

       }
   }
}




var mark = {
    firstName: 'marksd',
    bills: [77, 375, 110, 45],
    calTips: function(){
        this.tips = [];
        this.finlValues = [];


        for (var i =0; i < this.bills.length; i++);
        {
         var percentage;
         var bill =this.bills[i];
         if(bill < 100){
             percentage = 20/100;
         }else if (bill >= 100 && bill < 300){
             percentage = 10/100;
         }else{
             percentage = 25/100;
         }
         this.tips[i] = bill * percentage;
         this.finlValues[i] = bill + bill * percentage;

        }
    }
}
```

```
function calAverage(tips){
    var sum = 0;
    for(var i = 0; i<tips.length; i++){
        sum = sum + tips[i];
    }
    return sum / tips.length;

}

john.calTips();
mark.calTips();

john.average = calAverage(john.tips);
mark.average = calAverage(mark.tips);
console.log(john, mark);

if (john.average > mark.average) {
    console.log(firstName + 'john family pays higher tips'+ john.average);
}else if (mark.average > john.average){
    console.log('mark family pays higher');
}
```




```
//function coding challange 7 my drive cource


//iife function for private our code
//7
(function(){
    //1
    function Qestion(questions, answers, correct) {
        this.questions = questions;
        this.answers = answers;
        this.correct = correct;
    }

    Qestion.prototype.displayQuestion = function() {
    console.log(this.questions);
```

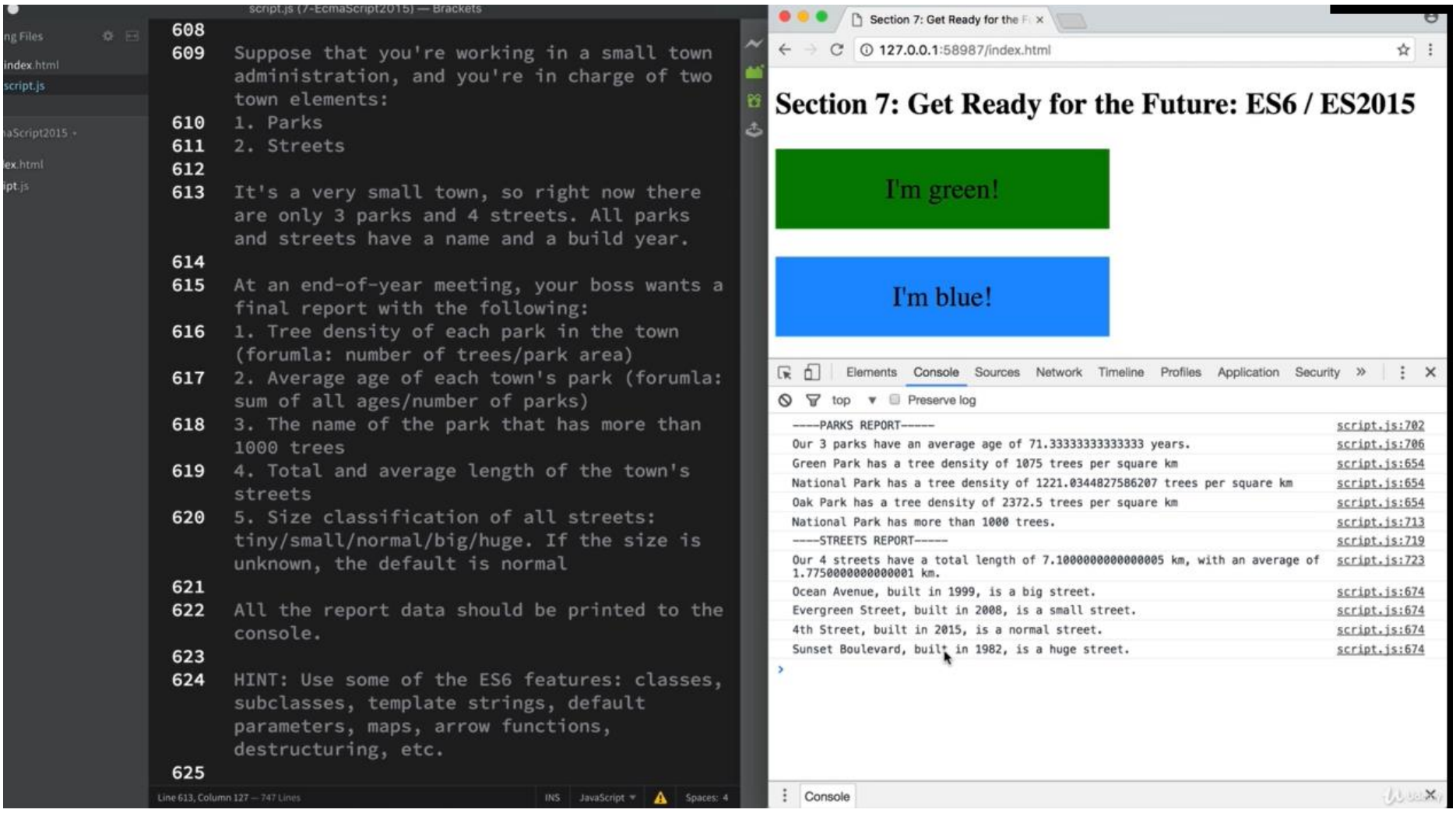
```
    //amra jani answer koto gula hote pare 2,3 etc so loop korbo
    for(var i=0; i<this.answers.length; i++){
        console.log(i + ':' + this.answers[i]);
     }
    }
    //6
    Qestion.prototype.checkAnswer = function(ans) {
        if (ans === this.correct) {
            console.log("correct answer!")
        }else{
            console.log("Wrong answer. Try again");
        }
    }
    //2
    //answer ekta variable tai ekta variable er modda amra array akare onek gula value rakte pari tai array
    var q1 = new Qestion('Js Every where?', ['yes', 'No'], 0);
    var q2 = new Qestion('Js create fun?', ['yes', 'No', 'both'], 2);
    var q3 = new Qestion('Js cource butuful?', ['yes', 'No'], 0);
    //3
    question = [q1, q2, q3];

    //4
    var n = Math.floor(Math.random() * question.length);
    question[n].displayQuestion();
    //5
    //Answer jano number hoi ai jonno parseInt //parseInt string k number a convert kore
    var answer = parseInt(prompt("Please secect the correct answer"));

    question[n].checkAnswer(answer);
    })()
```

```
330  --- Expert level ---
331
332  8. After you display the result, display the next random question, so that the game never ends (Hint:
     write a function for this and call it right after displaying the result)
333
334  9. Be careful: after Task 8, the game literally never ends. So include the option to quit the game if the
     user writes 'exit' instead of the answer. In this case, DON'T call the function from task 8.
335
336  10. Track the user's score to make the game more fun! So each time an answer is correct, add 1 point to
     the score (Hint: I'm going to use the power of closures for this, but you don't have to, just do this
     with the tools you feel more comfortable at this point).
337
338  11. Display the score in the console. Use yet another method for this.
339  */
340
```

```
//Expert level
//7
(function(){
    //1
    function Qestion(questions, answers, correct) {
        this.questions = questions;
        this.answers = answers;
        this.correct = correct;
    }

    Qestion.prototype.displayQuestion = function() {
    console.log(this.questions);
    //amra jani answer koto gula hote pare 2,3 etc so loop korbo
    for(var i=0; i<this.answers.length; i++){
        console.log(i + ':' + this.answers[i]);
     }
```

```
}
//6
Qestion.prototype.checkAnswer = function(ans, callback) {
    var sc;
    if (ans === this.correct) {
        console.log("correct answer!")
       sc = callback(true);
    }else{
        console.log("Wrong answer. Try again");
        sc = callback(false);
    }
    this.displayScore(sc);//displayScore ai khane pilam prototype er karone
}
Qestion.prototype.displayScore = function(score) {//sc pass hosse score er maddome


    console.log("Your current score is: " + score);
    console.log("----------------------")
}
//2
//answer ekta variable tai ekta variable er modda amra array akare onek gula value rakte pari tai array
var q1 = new Qestion('Js Every where?', ['yes', 'No'], 0);
var q2 = new Qestion('Js create fun?', ['yes', 'No', 'both'], 2);
var q3 = new Qestion('Js cource butuful?', ['yes', 'No'], 0);


//10
 question = [q1, q2, q3];
 function score() {
     var sc = 0;
     return function(correct){
         if (correct) {
             sc++;
         }
         return sc;
     }
 }
 var keepScore = score();//keepScore sc varible k use korbe closer er karone korte parbe

 //8
 function nextQuestion(){
  //4
  var n = Math.floor(Math.random() * question.length);
  question[n].displayQuestion();
  //5
  //Answer jano number hoi ai jonno parseInt //parseInt string k number a convert kore
  var answer = prompt("Please secect the correct answer");
//9
  if(answer !== 'exit') {//aita to cholte e thakbe tai exit dewa holo off korer jonno
      question[n].checkAnswer(parseInt(answer), keepScore);

      nextQuestion();//function er bitor e function call holo next next qus jano cholte thake
  }

}
nextQuestion();
 })()
```

ES6 codding challange




```
608
609  Suppose that you're working in a small town
     administration, and you're in charge of two
     town elements:
610  1. Parks
611  2. Streets
612
613  It's a very small town, so right now there
     are only 3 parks and 4 streets. All parks
     and streets have a name and a build year.
614
615  At an end-of-year meeting, your boss wants a
     final report with the following:
616  1. Tree density of each park in the town
     (forumla: number of trees/park area)
617  2. Average age of each town's park (forumla:
     sum of all ages/number of parks)
618  3. The name of the park that has more than
     1000 trees
619  4. Total and average length of the town's
     streets
620  5. Size classification of all streets:
     tiny/small/normal/big/huge. If the size is
     unknown, the default is normal
621
622  All the report data should be printed to the
     console.
623
624  HINT: Use some of the ES6 features: classes,
     subclasses, template strings, default
     parameters, maps, arrow functions,
     destructuring, etc.
625
```

```
----PARKS REPORT------                                                    script.js:702
Our 3 parks have an average age of 71.33333333333333 years.               script.js:706
Green Park has a tree density of 1075 trees per square km                 script.js:654
National Park has a tree density of 1221.0344827586207 trees per square km  script.js:654
Oak Park has a tree density of 2372.5 trees per square km                 script.js:654
National Park has more than 1000 trees.                                   script.js:713
----STREETS REPORT------                                                  script.js:719
Our 4 streets have a total length of 7.1000000000000005 km, with an average of
1.7750000000000001 km.                                                    script.js:723
Ocean Avenue, built in 1999, is a big street.                             script.js:674
Evergreen Street, built in 2008, is a small street.                       script.js:674
4th Street, built in 2015, is a normal street.                            script.js:674
Sunset Boulevard, built in 1982, is a huge street.                        script.js:674
```

```
//Coding challange with ES6 feature

class Element {
    constructor(name, buildYear){
        this.name = name;
        this.buildYear = buildYear;
    }
}

class Park extends Element {
    constructor(name, buildYear, area, numTress){
        super(name, buildYear);
        this.area = area;
        this.numTress = numTress;
    }
    treeDensity(){
        const density = this.numTress / this.area;
        console.log(`${this.name} has a tree density of ${density} trees per squire km`);
    }
}

class Street extends Element {
    constructor(name, buildYear, length, size = 3 ){
        super(name, buildYear);
        this.length = length;
        this.size = size;
    }
    classifyStreet() {
        const classificaton = new Map();
        classificaton.set(1, 'tiny');
        classificaton.set(2, 'small');
        classificaton.set(3, 'normal');
        classificaton.set(4, 'big');
        classificaton.set(5, 'huge');
        console.log(`${this.name}, build in ${this.buildYear}, is a ${classificaton.get(this.size)} street`);
```

```
    }
}

const allPark = [new Park('Green Park', 1987, 0.2, 215), new Park('National Park', 19894, 1.2, 21534), new Park('Blue Pa
rk', 1957, 2.2, 3215)];

const allStreets = [new Street('Ocean Avenue', 1990, 1.8, 4), new Street('Street Avenue', 1990, 1.2, 4), new Street('Rea
d streed Avenue', 2012, 2.2, 2), new Street('True street', 1999, 5.2, 5)];


function calc(arr) {
    const sum = arr.reduce((prev, cur, index) => prev + cur, 0); //reduce main array k change kore na (preValue,
currentValue)
    return [sum, sum / arr.length];

}

function reportPark(p) {
    console.log(`------------Park report-----------`);
    //Density
    p.forEach(el => el.treeDensity());

    //Average age
    const ages = p.map(el => new Date().getFullYear() - el.buildYear);
    const [totalAge, avgAge] = calc(ages);
    console.log(`Our ${p.length} parks have an average of ${avgAge} years.`);


    //Which park has more than 1000 trees
    const i = p.map(el => el.numTress).findIndex(el => el >= 1000);
    console.log(`${p[i].name} has more than 1000 trees.`);

}

function reportStrees(s) {
    console.log(`------------Street report-----------`);
    //Total and avg length of the town streets
    const [totalLength, avgLength] = calc(s.map(el => el.length)); //distructure kore niya ashlam totalLength & avgLengt
h
    console.log(`Our ${s.length} streets have a total length of ${totalLength} km, with an average of ${avgLength} km.`)
;

    //Classify size
    s.forEach(el => el.classifyStreet());//size=3 kora ai jonno classifyStreet() bitor diya kisu pass kora hoi nai

}

reportPark(allPark);
reportStrees(allStreets);



// output:
// ------------Park report-----------
//  Green Park has a tree density of 1075 trees per squire km
//  National Park has a tree density of 17945 trees per squire km
//  Blue Park has a tree density of 1461.3636363636363 trees per squire km
//  Our 3 parks have an average of -5925 years.
//  National Park has more than 1000 trees.
//  ------------Street report-----------
//  Our 4 streets have a total length of 10.4 km, with an average of 2.6 km.
//  Ocean Avenue, build in 1990, is a big street
//  Street Avenue, build in 1990, is a big street
//  Read streed Avenue, build in 2012, is a small street
//  True street, build in 1999, is a big street
```