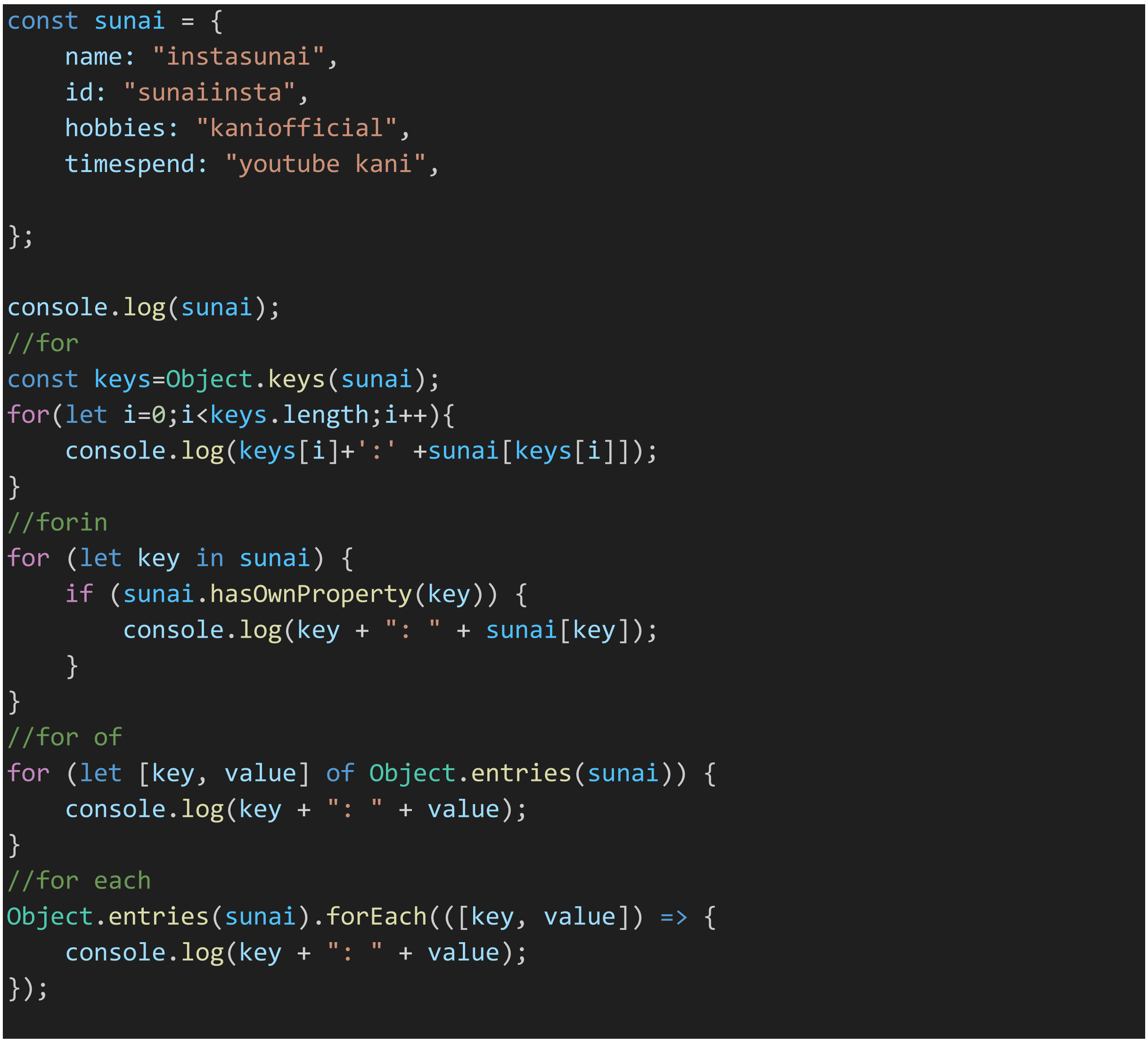
1.For the given Json iterate over all for loops (for,forin,forof,foreach)

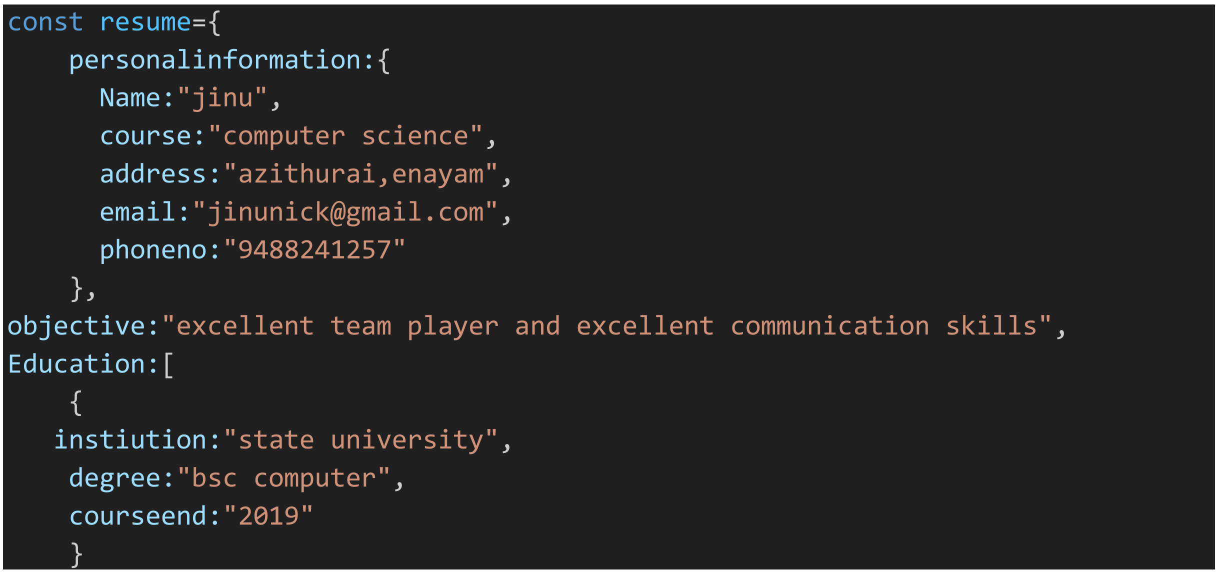


```
const sunai = {
    name: "instasunai",
    id: "sunaiinsta",
    hobbies: "kaniofficial",
    timespend: "youtube kani",

};

console.log(sunai);
//for
const keys=Object.keys(sunai);
for(let i=0;i<keys.length;i++){
    console.log(keys[i]+':' +sunai[keys[i]]);
}
//forin
for (let key in sunai) {
    if (sunai.hasOwnProperty(key)) {
        console.log(key + ": " + sunai[key]);
    }
}
//for of
for (let [key, value] of Object.entries(sunai)) {
    console.log(key + ": " + value);
}
//for each
Object.entries(sunai).forEach(([key, value]) => {
    console.log(key + ": " + value);
});
```

2.Create your own resume data in Json format



```
const resume={
    personalinformation:{
      Name:"jinu",
      course:"computer science",
      address:"azithurai,enayam",
      email:"jinunick@gmail.com",
      phoneno:"9488241257"
    },
objective:"excellent team player and excellent communication skills",
Education:[
    {
   instiution:"state university",
    degree:"bsc computer",
    courseend:"2019"
    }
```

```
],
workexperience:[
    {
        company:"altruist technologies",
        position:"customer service supervisor"
    }
    ],
    skills:[
    "javascript",
     "html",
     "css"
],
certifications:[
    "desktop publishing",
    "learning html",
],

};
console.log("personal information:",resume.personalinformation);
console.log("objective:",resume.objective);
console.log("education:",resume.Education);
console.log("work experience:",resume.workexperience);
console.log("skills:",resume.skills)
console.log("certifications:",resume.certifications);
```

Result:

Personal information:

Name:jinu

Address:azithurai,enayam

Course:computer science

Email:jinunick@gmail.com

Objective:

Excellent team player and excellent communication skills

Education:

Institution: Madras university

Course: bsc computer

Course end:2019

Work experience:

Company: altruist technologies

Position: customer service supervisor

Skills:

Javascript,html,css

Certifications:

Desktop publishing, Learning html

3)Read about the difference between window, screen and document in javascript

In JavaScript, the terms `window`, `screen`, and `document` refer to different objects, each representing different aspects of a web page and its environment. Understanding the distinctions among them is crucial for effective web development. Here's a brief overview of each:

1. `window`

- **Scope:** The `window` object represents a browser window or a frame within a browser window. It is the top-level object in the browser environment, meaning it's at the top of the scope chain.
- **Properties and Methods:** It includes properties and methods to control the browser window, like `window.innerWidth`, `window.innerHeight`, `window.open()`, `window.close()`, etc. It also includes the `document` object.
- **Global JavaScript Environment:** In browsers, the global JavaScript environment is the `window`. So, variables and functions declared globally are properties of the `window` object.
- **Event Handling:** You can attach event listeners to the `window` object for events that affect the entire browser window, like load or resize events.

2. `screen`

- **Purpose:** The `screen` object provides information about the user's screen, such as its resolution. It's mainly used for understanding the physical display characteristics.
- **Properties:** Includes properties like `screen.width`, `screen.height`, `screen.availWidth`, `screen.availHeight`, etc. These properties give information about the screen's total width and height and the width and height available to the browser window.
- **Use Cases:** Commonly used in responsive web design to adapt the layout to different screen sizes or to optimize resources and content for different screen resolutions.

3. `document`

- **Role:** The `document` object represents the HTML document loaded in that window and is a part of the Document Object Model (DOM). It's the root node of the HTML document.
- **Manipulation:** It allows you to manipulate the content, structure, and style of the website. For example, `document.getElementById()`, `document.createElement()`, `document.querySelector()`, etc., are used to interact with the HTML elements.
- **Events:** Event listeners can be added to the `document` object for events that are specific to the entire webpage, like DOMContentLoaded.

4)Code keta practice:

1)Find the minimum value among 5 numbers

Const array=[1,2,3,4,5]

Ans:

const array = [1, 2, 3, 4, 5];

const minValue = Math.min(...array);

console.log(minValue);

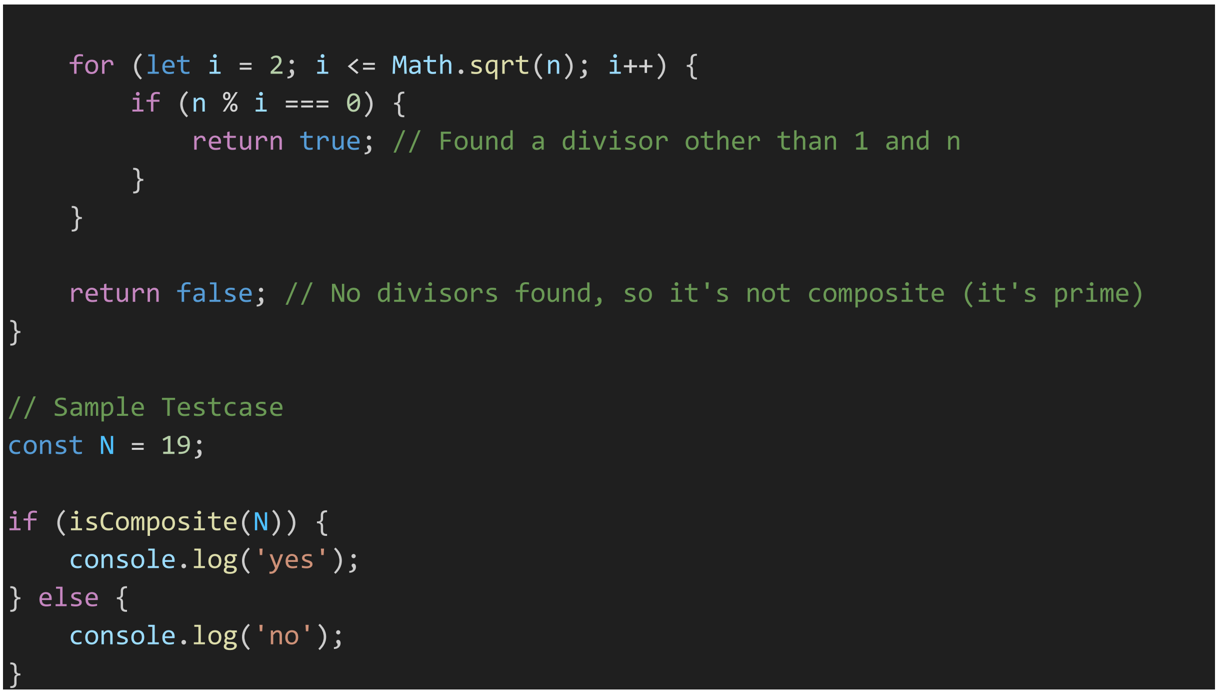
Result:

1

2) Given a number N, print 'yes' if it is composite else print 'no'.
Sample Testcase :
INPUT
123
OUTPUT
yes

Ans:

```
function isComposite(n) {
    if (n <= 1) {
        return false; // Numbers less than or equal to 1 are neither prime nor
composite
    }
```



```
    for (let i = 2; i <= Math.sqrt(n); i++) {
        if (n % i === 0) {
            return true; // Found a divisor other than 1 and n
        }
    }

    return false; // No divisors found, so it's not composite (it's prime)
}

// Sample Testcase
const N = 19;

if (isComposite(N)) {
    console.log('yes');
} else {
    console.log('no');
}
```

Result:

No

3)Codeketa practice

1)Write a code to get the input and print it 5 times

Ans:

```
N = [1];


for (let j = 0; j < 5; j++) { // Outer loop to repeat the process 5 times
   for (let i = 0; i < N.length; i++) { // Inner loop to go through array N
      console.log(N[i]);
   }
}
```

Result:

1 1 1 1 1

2)Write a code to get an integer N and print the values from N to 1

Ans:

```
const N=[1,2,3,4,5];
console.log(N);
Array=N.reverse();
console.log(Array);
```

Result:

5 4 3 2 1

3)Given 2 numbers N,M .print ‘yes’ if their product is a perfect square else print ‘no’.

Ans:

```
let N = 9;
let M = 8;

let product = N * M;
let sqrt = Math.sqrt(product);

if (sqrt === Math.floor(sqrt)) {
    console.log("yes");
} else {
    console.log("no");
}
```

Result:

No

4)You are provided with a number ‘n’.your task is to tell whether that number is saturated.

A Saturated number is a number which is made by exactly two digits.

Ans:

```
function isSaturated(numberArray) {
    let uniqueDigits = new Set(numberArray);

    // A saturated number has exactly two unique digits
    return uniqueDigits.size === 2;
}

const n = [2, 3, 4];
console.log(isSaturated(n));
```

Result:

False