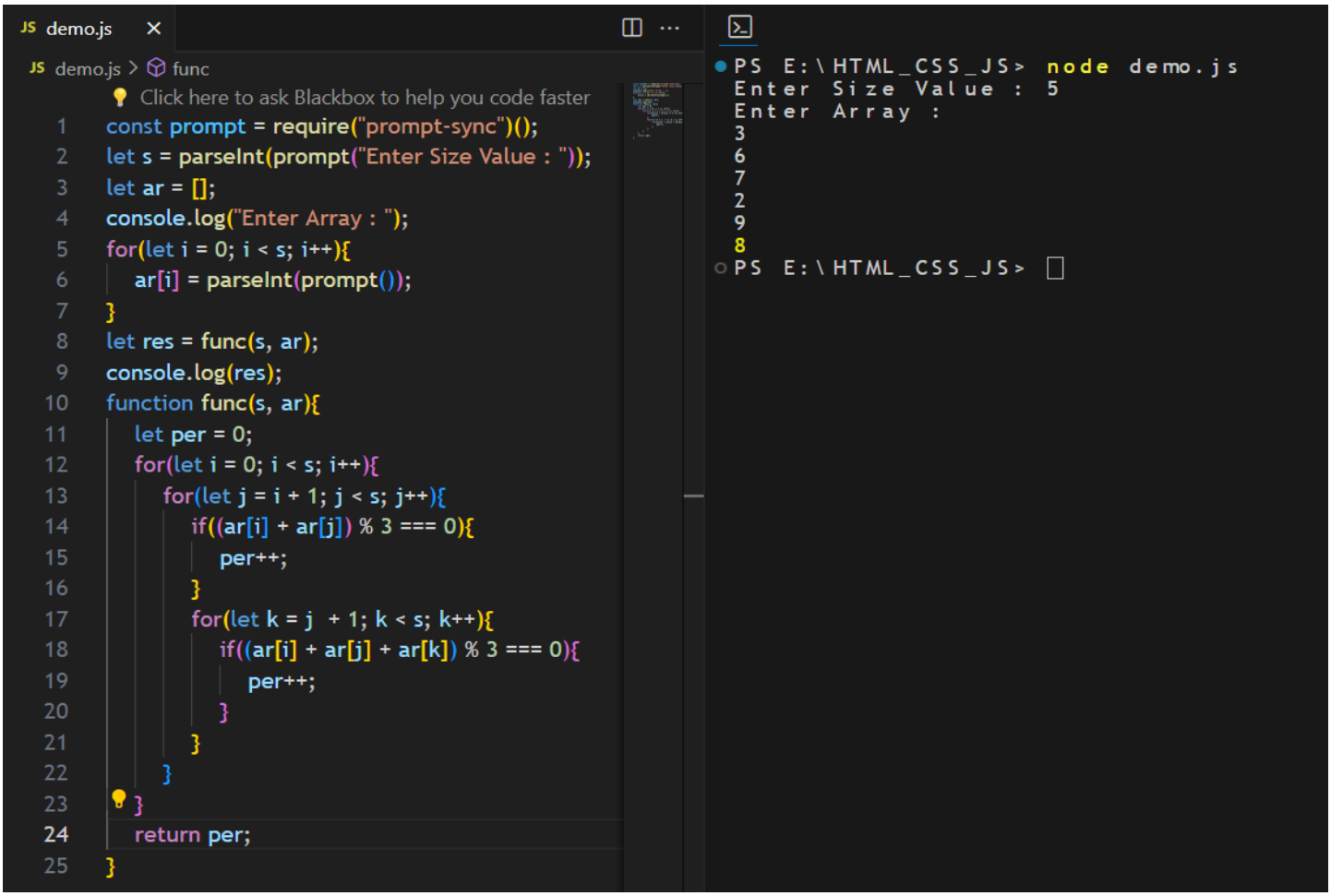
Question01:

In a class A perfect team is the team where sum of marks of students in the team is divisible by 3. It can be of two or three students Only. You have given an array representing the marks of students in class. you have to find how many perfect teams can be formed. Input: N = 5 Ar: [3, 6, 7, 2, 9] Output: 8



```
demo.js
Click here to ask Blackbox to help you code faster
const prompt = require("prompt-sync")();
let s = parseInt(prompt("Enter Size Value : "));
let ar = [];
console.log("Enter Array : ");
for(let i = 0; i < s; i++){
    ar[i] = parseInt(prompt());
}
let res = func(s, ar);
console.log(res);
function func(s, ar){
    let per = 0;
    for(let i = 0; i < s; i++){
        for(let j = i + 1; j < s; j++){
            if((ar[i] + ar[j]) % 3 === 0){
                per++;
            }
            for(let k = j  + 1; k < s; k++){
                if((ar[i] + ar[j] + ar[k]) % 3 === 0){
                    per++;
                }
            }
        }
    }
    return per;
}
```

```
PS E:\HTML_CSS_JS> node demo.js
Enter Size Value : 5
Enter Array :
3
6
7
2
9
8
PS E:\HTML_CSS_JS>
```

Question02:

IP: A = 2 B = 4 OP: 11 IP: A = 2 B = 1 OP: 0

```
demo.js
Click here to ask Blackbox to help you code faster
const prompt = require("prompt-sync")();
function findN(K, M) {
    if (K === 1 && M === 1) {  return 1;  }
    if (K === 1 && M !== 1) {  return 0;  }
    let N = K * Math.pow(10, M - 2) - 1;
    while (N > 0) {
        let currentPosition = getPosition(N, K);
        if (currentPosition === M) {
            return N;
        } else if (currentPosition < M) {
            N++;
        } else {
            N--;
        }
    }
    return 0;
}
function getPosition(N, K) {
    let lexicographicalOrder = getLexicographicalOrder(N);
    let position = lexicographicalOrder.indexOf(K) + 1;
    return position;
}
function getLexicographicalOrder(N) {
    return Array.from({ length: N }, (_, index) => index + 1).sort((a, b) => a.toString().localeCompare(b.toString()));
}
console.log(findN(2, 4));
console.log(findN(2, 1));
```

```
PS E:\HTML_CSS_JS> node demo.js
11
0
PS E:\HTML_CSS_JS>
```

Question03:

an integer arrays nums is provided and a target is provided. return true if any element has occurred more than target times else return false.

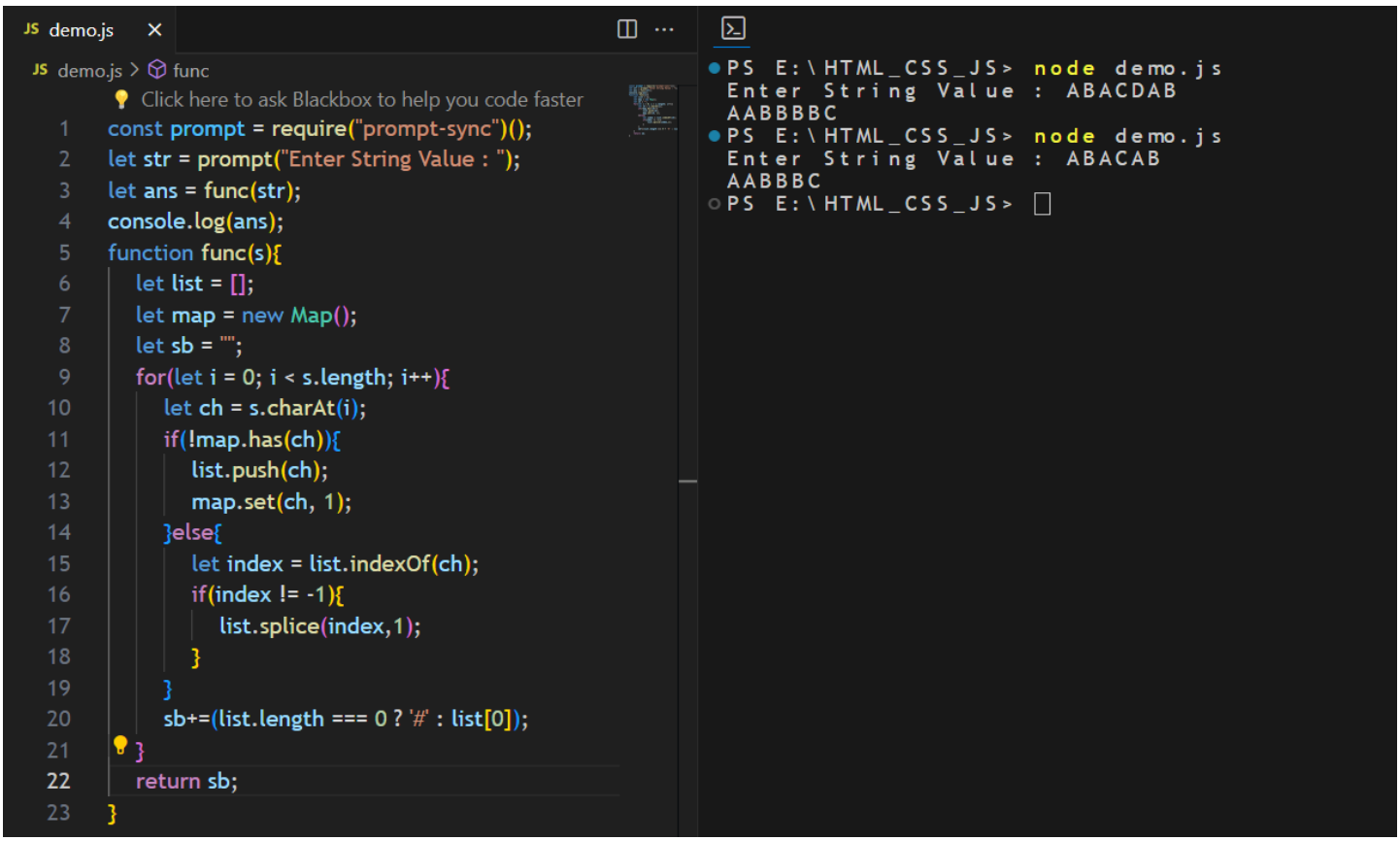
IP: [9 5 3 2 1 5 1 4 3 5 4 6 5] Target = 3 OP: true

```
demo.js
Click here to ask Blackbox to help you code faster
const prompt = require("prompt-sync")();
function func(nums, target) {
    let count = new Map();
    for (let num of nums) {
        count.set(num, (count.get(num) || 0) + 1);
    }
    for (let [num, c] of count) {
        if (c > target) {
            return true;
        }
    }
    return false;
}
let s = parseInt(prompt("Enter Size: "));
let ar = [];
console.log("Enter Array: ");
for (let i = 0; i < s; i++) {
    ar[i] = parseInt(prompt());
}
let n = parseInt(prompt("Enter Target Value: "));
let OP = func(ar, n);
console.log(OP);
```

```
PS E:\HTML_CSS_JS> node demo.js
Enter Size: 6
Enter Array:
10
8
10
4
10
1
Enter Target Value: 2
true
PS E:\HTML_CSS_JS> node demo.js
Enter Size: 13
Enter Array:
9
5
3
2
1
5
1
4
3
5
4
6
5
Enter Target Value: 5
false
PS E:\HTML_CSS_JS>
PS E:\HTML_CSS_JS>
PS E:\HTML_CSS_JS>
PS E:\HTML_CSS_JS>
```

Question04:

First non-repeating character in a stream of characters. Given a string A denoting a stream of lowercase alphabets.

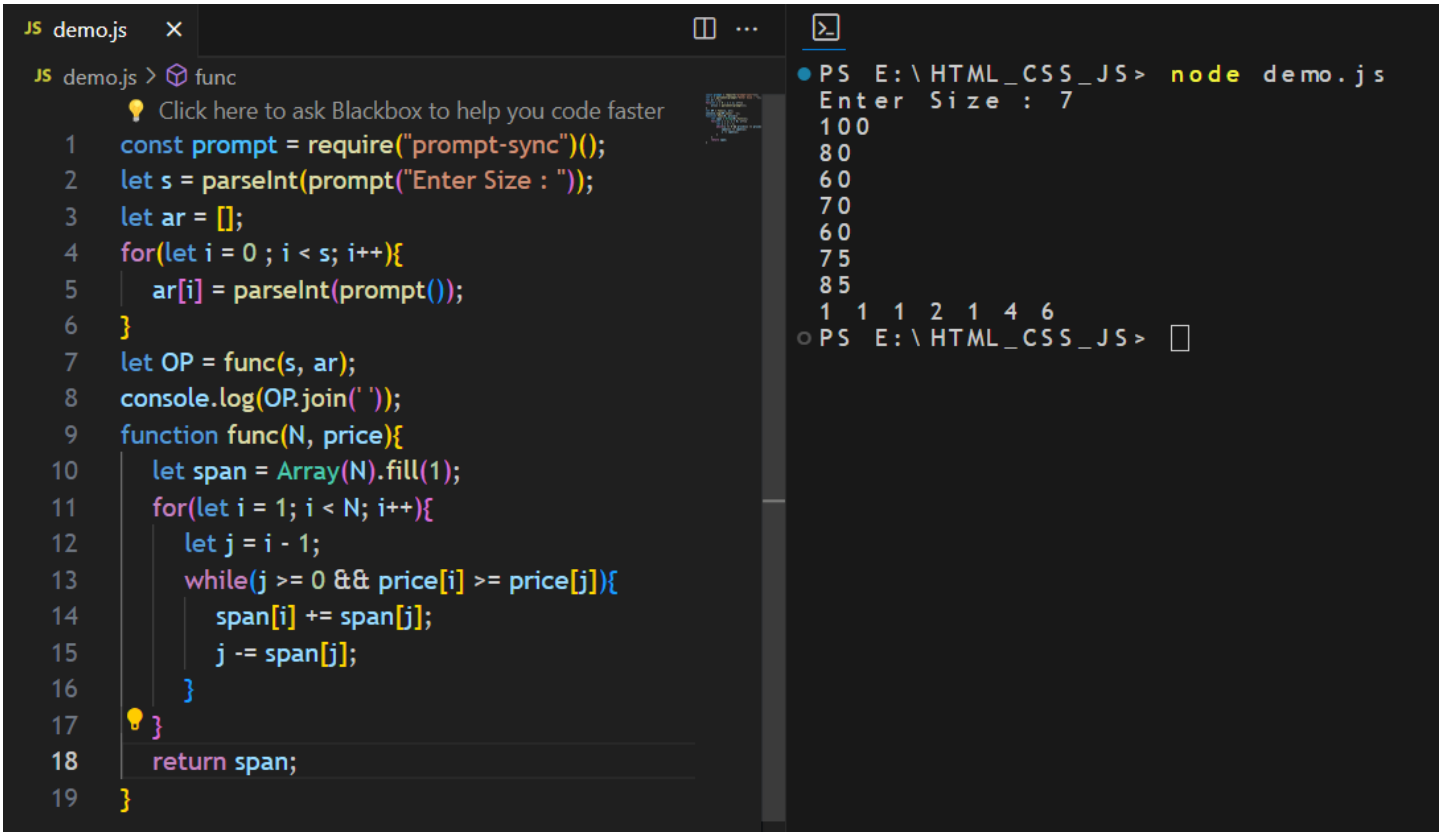


```
JS demo.js
JS demo.js > func
Click here to ask Blackbox to help you code faster
1  const prompt = require("prompt-sync")();
2  let str = prompt("Enter String Value : ");
3  let ans = func(str);
4  console.log(ans);
5  function func(s){
6      let list = [];
7      let map = new Map();
8      let sb = "";
9      for(let i = 0; i < s.length; i++){
10         let ch = s.charAt(i);
11         if(!map.has(ch)){
12             list.push(ch);
13             map.set(ch, 1);
14         }else{
15             let index = list.indexOf(ch);
16             if(index != -1){
17                 list.splice(index,1);
18             }
19         }
20         sb+=(list.length === 0 ? '#' : list[0]);
21     }
22     return sb;
23 }
```

```
PS E:\HTML_CSS_JS> node demo.js
Enter String Value : ABACDAB
AABBBBC
PS E:\HTML_CSS_JS> node demo.js
Enter String Value : ABACAB
AABBBC
PS E:\HTML_CSS_JS>
```

Question05:

IP: N = 7, price[] = [100 80 60 70 60 75 85] OP: 1 1 1 2 1 4 6



```
JS demo.js
JS demo.js > func
Click here to ask Blackbox to help you code faster
1  const prompt = require("prompt-sync")();
2  let s = parseInt(prompt("Enter Size : "));
3  let ar = [];
4  for(let i = 0 ; i < s; i++){
5      ar[i] = parseInt(prompt());
6  }
7  let OP = func(s, ar);
8  console.log(OP.join(' '));
9  function func(N, price){
10     let span = Array(N).fill(1);
11     for(let i = 1; i < N; i++){
12         let j = i - 1;
13         while(j >= 0 && price[i] >= price[j]){
14             span[i] += span[j];
15             j -= span[j];
16         }
17     }
18     return span;
19 }
```

```
PS E:\HTML_CSS_JS> node demo.js
Enter Size : 7
100
80
60
70
60
75
85
1 1 1 2 1 4 6
PS E:\HTML_CSS_JS>
```

Question06:

You are given an array Containing n-1 distinct numbers from range [1, n]. There is one element missing from the range[1,n] in the array. Find that missing Number and return it. Solve it without using any extra array.