Front-End Developer (React) Test Vaishali ... > vaishali.kapse@wipro.com

You can view this report online at : https://www.hackerrank.com/x/tests/1526462/candidates/49975169/report

Full Name:	Vaishali Kapse
Email:	vaishali.kapse@wipro.com
Test Name:	**Front-End Developer (React) Test Vaishali Kapse**
Taken On:	7 Feb 2023 15:34:00 IST
Time Taken:	59 min 52 sec/ 60 min
Work Experience:	< 1 years
Linkedin:	https://www.linkedin.com/in/vaishali-kapse-a87b4a1b6
Invited by:	Kavitha
Invited on:	6 Feb 2023 21:53:35 IST
Skills Score:	CSS 10/15 JavaScript (Basic) 0/50 Problem Solving (Basic) 5.5/50 React (Basic) 0/50
Tags Score:	Algorithms 5.5/50 Bit Manipulation 5.5/50 Box Model 0/5 CSS 5/5 CSS3 5/10 Closures 0/50 ES6 0/50 Easy 10.5/155 Front-End Development 5/5 Front-End Frameworks 0/50 HTML5 5/5 Interviewer Guidelines 5.5/50 JavaScript 0/50 Medium 5/10 Problem Solving 5.5/50 React 0/50
Candidate Tags:	frontend-react,frontend-test

9.4%

15/165

scored in **Front-End Developer (React) Test Vaishali Kapse** in 59 min 52 sec on 7 Feb 2023 15:34:00 IST

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	Powers of 2 > **Coding**	30 min 39 sec	5.5/ 50	✓
Q2	JavaScript: Inventory List > **Coding**	3 min 16 sec	0/ 50	!
Q3	Field Creation > **Multiple Choice**	1 min 8 sec	5/ 5	✓
Q4	CSS nth Selector > **Multiple Choice**	4 min 27 sec	5/ 5	✓
Q5	CSS: Box Model with Box Sizing > **Multiple Choice**	1 min 10 sec	0/ 5	☒
Q6	React: Autocorrection App > **Front-end Developer**	19 min 31 sec	0/ 50	☒

QUESTION 1

✓

Correct Answer

Score 5.5

Powers of 2 › Coding

Easy | Bit Manipulation | Algorithms | Problem Solving | Interviewer Guidelines

QUESTION DESCRIPTION

Given an array of integers, determine whether each is a power of *2,* where powers of *2* are *[1, 2, 4, 8, 16, 32, 64,…].* For each integer evaluated, append to an array a value of *1* if the number is a power of *2* or *0* otherwise.

Example

arr = [1, 3, 8, 12, 16]

$1 = 2^0$, $8 = 2^3$ and $16 = 2^4$. The return array is *[1, 0, 1, 0, 1].*

Function Description

Complete the function *isPower* in the editor below.

isPower has the following parameter(s):

int arr[n]: an array of integers

Returns

int[n]: array of binary integers where each index *i* contains a *1* if *arr[i]* is a power of *2* or a *0* if it is not

Constraints

- $1 \le n \le 100$
- $0 \le arr[i] \le 5 \times 10^7$

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array *arr*.

Each of the next *n* lines contains an integer *arr[i]* where $0 \le i < n$.

▼ Sample Case 0

Sample Input 0

```
STDIN     Function
```

```
-----      --------
3       →  arr[] size n = 3
2       →  arr = [2, 3, 4]
3
4
```

Sample Output 0

```
1
0
1
```

Explanation 0

Evaluate the following *n = 3* integers in *arr = [2, 3, 4]*:

0. *arr[0] = 2 =* 2^1, so the answer for this index is *1*.
1. *arr[1] = 3* is not a power of *2*, so the answer for this index is *0*.
2. *arr[2] = 4 =* 2^2, so the answer for this index is *1*.

Return the array *[1, 0, 1]* as the answer.

▼ Sample Case 1

Sample Input 1

```
STDIN         Function
-----         --------
3        →    arr[] size n = 3
1024     →    arr = [1024, 2048, 1048576]
2048
1048576
```

Sample Output 1

```
1
1
1
```

Explanation 1

Evaluate the following *n = 3* integers in *arr = [1024, 2048, 1048576]*:

0. *arr[0] = 1024 =* 2^{10}, so the answer for this index is *1*.
1. *arr[1] = 2048 =* 2^{11}, so the answer for this index is *1*.
2. *arr[2] = 1048576 =* 2^{20}, so the answer for this index is *1*.

Return the array *[1, 1, 1]* as the answer.

INTERVIEWER GUIDELINES

▼ Hint 1

Think about binary representation of the numbers in the array. How many bits are set in a power of 2? (Answer: 1)

▼ Hint 2

Think about bitwise operations. What logical operation against what pattern of bits will return 'False' if a number is a power of 2? (Answer: logical AND with a number that is 1 bit shorter having all bits set, e.g. 8 = 1000, 1000 & 111 = 0.

▼ Solution

Concepts covered:

This problem tests the candidate's knowledge of bitwise operations.

Optimal Solution:

If a number X is a power of 2, then the result of the bitwise operation X & (X-1) will be 0. The only special case is when X = 0, which is not a power of two but the operation will result 0. So, we handle this case separately.

The time complexity is O(N), where N is the total number of elements in the input array.

```
def isPower(arr):
    ans = []
    for x in arr:
        if x == 0:
            ans.append(0)
        elif (x & (x-1)) == 0:
            ans.append(1)
        else:
            ans.append(0)
    return ans
```

Brute Force Approach:

For each number, keep dividing the number by 2 until it is odd. If the resulting odd number is 1, then the number is a power of 2.

The time complexity of this approach is O(NlogN) where N is the total number of elements in the input array.

Error Handling:

1. A zero value is a special case that must be handled separately.

▼ Complexity Analysis

Time Complexity - O(N).

The time complexity of the bitwise operation to check if a number is a power of two or not is O(1) so the total time complexity is O(N) where N is the total number of elements in the input array.

Space Complexity - O(N).

Since we have to build the answer array of size equal to the size of the input array, the space complexity is O(N).

CANDIDATE ANSWER

Language used: **JavaScript (Node.js)**

```
/*
 * Complete the 'isPower' function below.
 *
 * The function is expected to return an INTEGER_ARRAY.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

function isPower(arr) {
    // Write your code here
        let a=[]
        for(let i=0;i<arr.length;i++){
            if(arr[i]=0){
                a.push(0);
            }
            else if(Number.isInteger(Math.sqrt(arr[i]))){
                a.push(1);
            }
            else{
                a.push(0);
            }
        }
        return a;

}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED

TestCase 0	Easy	Sample case	⊗ Wrong Answer	0	0.0513 sec	38.4 KB
TestCase 1	Easy	Sample case	✓ Success	0.5	0.049 sec	38.4 KB
TestCase 2	Medium	Sample case	⊗ Wrong Answer	0	0.0537 sec	38.5 KB
TestCase 3	Medium	Sample case	⊗ Wrong Answer	0	0.0531 sec	38.4 KB
TestCase 4	Medium	Hidden case	✓ Success	5	0.0577 sec	38.5 KB
TestCase 5	Medium	Hidden case	⊗ Wrong Answer	0	0.0702 sec	38.5 KB
TestCase 6	Hard	Hidden case	⊗ Wrong Answer	0	0.0528 sec	38.2 KB
TestCase 7	Hard	Hidden case	⊗ Wrong Answer	0	0.0477 sec	38.2 KB
TestCase 8	Hard	Hidden case	⊗ Wrong Answer	0	0.0674 sec	38.6 KB

No Comments

QUESTION 2

! Needs Review

Score 0

JavaScript: Inventory List › Coding

ES6 Easy Closures JavaScript

QUESTION DESCRIPTION

In this challenge, the task is to implement a function *inventoryList* such that*:*

- it maintains the collection of all item names existing in an inventory, where each item is uniquely identified by a name.
- returns a new object, with three methods:
  - add(name) - The string *name* parameter is passed, and it is added to the collection. It is guaranteed that at any time, if an item is in the collection, then no other item with the same name will be added to the collection.
  - remove(name) - The string *name* parameter is passed, and this item is removed from the collection if it exists. If it does not exist, nothing happens.
  - getList() - This returns an array of names of items added so far. The names are returned in the order the corresponding items were added.

Your implementation of the function will be tested by a stubbed code on several input files. Each input file contains parameters for the functions call. The functions will be called with those parameters, and the result of their executions will be printed to the standard output by the provided code. The stubbed code joins the strings returned by *getList* function by a comma and prints to the standard output. If *getList* returns an empty array, the stubbed code prints 'No Items'.

Constraints:

- The size of the collection will not exceed 10 at any point.
- All names passed to add(name) and remove(name) are non-empty.

▼ Input Format For Custom Testing

In the first line, there is an integer, *n*, denoting the number of operations to be performed.
Each line *i* of the *n* subsequent lines (where *0 ≤ i < n*) contains space-separated strings such that the first of them is a function name, and the remaining ones, if any, are parameters for that function.

▼ Sample Case 0

Sample Input For Custom Testing

```
5
add Shirt
add Trouser
getList
```

```
remove Shirt
getList
```

Sample Output

```
Shirt,Trouser
Trouser
```

Explanation

Items 'Shirt' and 'Trouser' are added by the *add* function. Then, *getList* is called and the result is printed. Item 'Shirt' is removed by calling the *remove* function. Finally, *getList* is called and the result is printed.

▼ Sample Case 1

Sample Input For Custom Testing

```
3
add Shirt
remove Trouser
getList
```

Sample Output

```
Shirt
```

Explanation

Item 'Shirt' is added by the *add* function. Then, *remove* is called with 'Trouser', but since it does not exist, nothing happens. Finally, *getList* is called and the result is printed.

INTERVIEWER GUIDELINES

tester's solution:

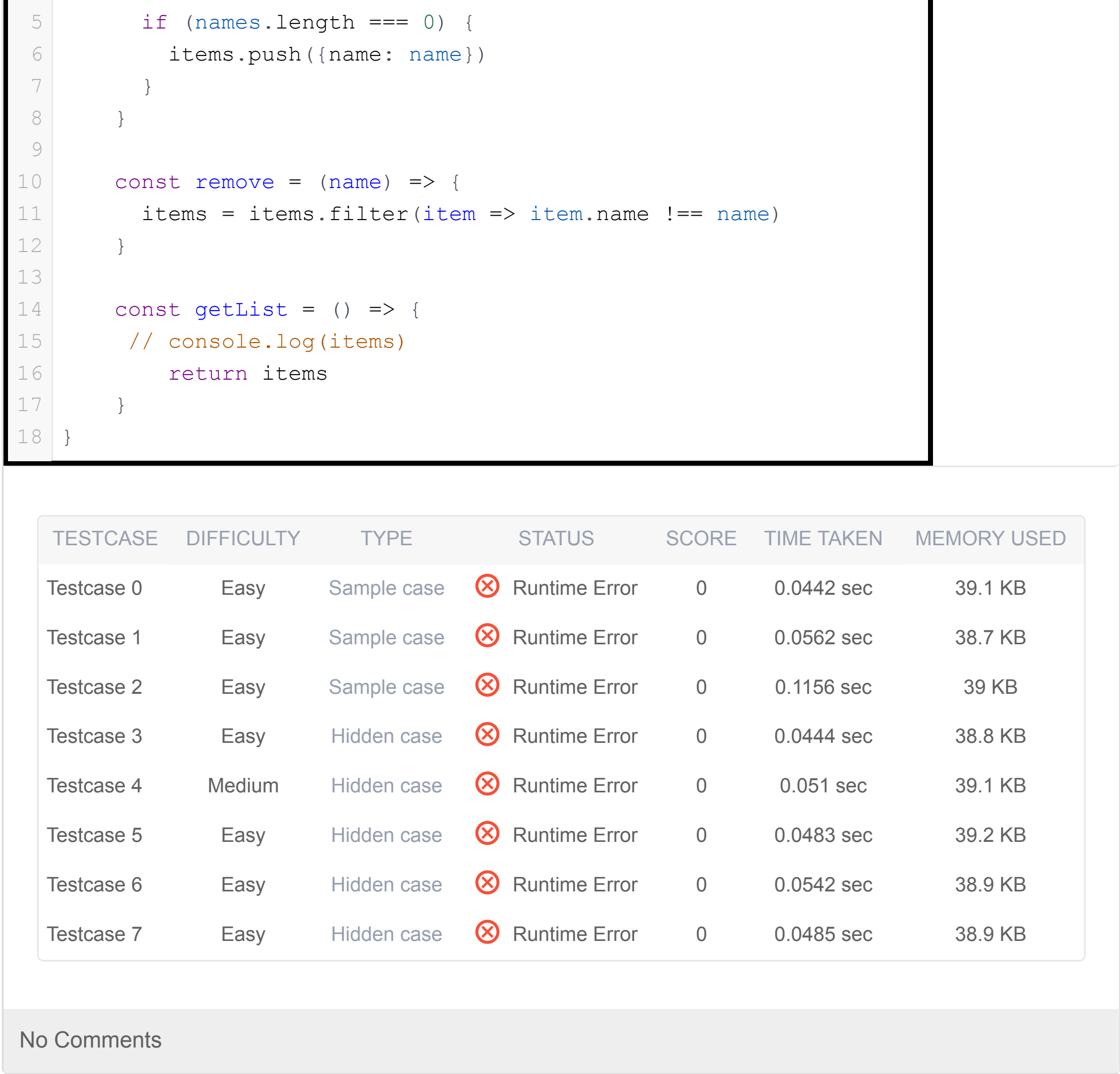
```
const assert = require('assert');
function inventoryList() {
  const names = [];
  return {
    add: (name) => {
      assert(name !== '');
      assert(!names.includes(name));
      assert(names.length <= 10);
      names.push(name);
    },
    remove: (name) => {
      const idx = names.indexOf(name);
      if (idx !== -1) {
        names.splice(idx, 1);
      }
    },
    getList: () => [...names],
  };
}
```

CANDIDATE ANSWER

The candidate did not manually submit any code. The last compiled version has been auto-submitted and the score you see below is for the auto-submitted version.

Language used: **JavaScript (Node.js)**

```
function inventoryList() {
  // write your code here
      const add = (name) => {
      const names = items.filter(item => item.name === name)
```



```
      if (names.length === 0) {
        items.push({name: name})
      }
    }

    const remove = (name) => {
      items = items.filter(item => item.name !== name)
    }

    const getList = () => {
     // console.log(items)
        return items
    }
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	⊗ Runtime Error	0	0.0442 sec	39.1 KB
Testcase 1	Easy	Sample case	⊗ Runtime Error	0	0.0562 sec	38.7 KB
Testcase 2	Easy	Sample case	⊗ Runtime Error	0	0.1156 sec	39 KB
Testcase 3	Easy	Hidden case	⊗ Runtime Error	0	0.0444 sec	38.8 KB
Testcase 4	Medium	Hidden case	⊗ Runtime Error	0	0.051 sec	39.1 KB
Testcase 5	Easy	Hidden case	⊗ Runtime Error	0	0.0483 sec	39.2 KB
Testcase 6	Easy	Hidden case	⊗ Runtime Error	0	0.0542 sec	38.9 KB
Testcase 7	Easy	Hidden case	⊗ Runtime Error	0	0.0485 sec	38.9 KB

No Comments

QUESTION 3

✓ Correct Answer

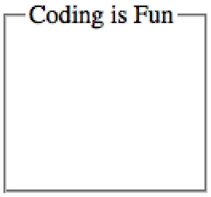
Score 5

Field Creation › Multiple Choice

Easy | HTML5 | CSS | Front-End Development

QUESTION DESCRIPTION

Which code produces the following? Width and height are 100px.



CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

✓ ● <fieldSet style="height:100px; width:100px"> <Legend>Coding is Fun</Legend> </fieldSet>

○ <fieldSet style="height:100px; width:100px"> <Header>Coding is Fun</Header> </fieldSet>

○ <fieldSet style="height:100px; width:100px"> <Caption>Coding is Fun</Caption> </fieldSet>

○ <fieldSet style="height:100px; width:200px"> <Legend>Coding is Fun</Legend> </fieldSet>

No Comments

QUESTION 4

✓

Correct Answer

Score 5

CSS nth Selector › Multiple Choice CSS3 Medium

QUESTION DESCRIPTION

What does this CSS code do?

```
p:nth-child(2n+2) {
    color: #ccc;
}
```

INTERVIEWER GUIDELINES

Usage of CSS nth selector property

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

○ Selects the 4th, 6th, 8th, etc. <p> tags

○ Selects all the even <p> tags

✓ ● Selects the 2nd, 4th, 6th, 8th, etc. (even) <p> tags starting from 2

○ None of the above

No Comments

QUESTION 5

Wrong Answer

Score 0

CSS: Box Model with Box Sizing › Multiple Choice

CSS3 | Medium | Box Model

QUESTION DESCRIPTION

Select all the statements that are correct regarding the following snippet, which makes use of the 'box-sizing' property.

```
<style>
  * {
    box-sizing: border-box;
  }
  p {
    width: 300px;
    height: 100px;
    padding: 10px;
    border: 10px solid red;
    margin: 10px;
    display: inline-block;
  }
</style>
<body>
  <p>This is a paragraph</p>
</body>
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

✓ ○ The total width of the the paragraph <p> element is 320px.

● The total width of the the paragraph <p> element is 360px.

✓ ○ The total height of the the paragraph <p> element is 120px.

● The total height of the the paragraph <p> element is 160px.

No Comments

QUESTION 6

Wrong Answer

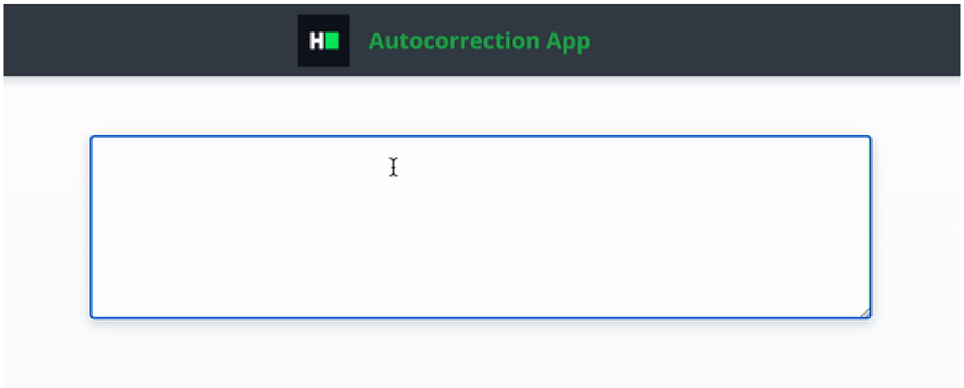
Score 0

React: Autocorrection App › Front-end Developer

Easy | Front-End Frameworks | React

QUESTION DESCRIPTION

Create a basic autocorrection application per the requirements below. The finished application must pass all of the unit tests.



Complete the implementation of *src/components/AutocorrectTextarea.js* according to the following requirements:

- AutocorrectTextarea is a component that takes a corrections Object that maps strings to their corrections. For example, the object below denotes that 'really' is a correction for 'realy', and 'weird' is a correction of 'wierd':

  ```
  const corrections = {
    'realy': 'really',
    'wierd': 'weird',
  };
  ```

- Assume that no value of the corrections object appears as the property in the corrections object.
- AutocorrectTextarea renders a textarea element and lets users write text in it.
- Assume that the text consists only of words separated by a single space character.
- Once a space character is typed, the word preceding it is considered to be complete and must be autocorrected according to the corrections object if a correction exists.

Initially, the file is filled with boilerplate code. Note the following:

- The textarea element must have `data-testid="textarea"`.

Please note that the component has these data-testid attributes for test cases, and certain classes and ids for rendering purposes. You should not change them.

INTERVIEWER GUIDELINES

▼ src/components/AutocorrectTextarea.js

```
import React, {useState} from 'react';

const AutocorrectTextarea = (props) => {
    const [text, setText] = useState('');

    const handleChange = (event) => {
        const {corrections} = props;
        const {value} = event.target;
        if (value.length && value[value.length - 1] === ' ') {
            const words = value.split(' ');
            words.pop();
            const lastWord = words.pop();
            setText([...words, (corrections[lastWord] ||
lastWord)].join(' ') + ' ');
        } else {
            setText(value);
        }
    }


    return (
        <div className="text-center">
            <textarea data-testid="textarea" rows={10} cols={80}
className="card" value={text}
        </div>
    );

}


export default AutocorrectTextarea;
```

CANDIDATE SUBMISSION

TESTCASE	TEST FILE	STATUS	SCORE

TESTCASE	TEST FILE	STATUS	SCORE
Test without removing characters	junit.xml	⊗ Failed	0 / 25.0
Test with removing characters	junit.xml	⊗ Failed	0 / 25.0

View candidate code

Review logs: output log

No Comments

PDF generated at: 7 Feb 2023 17:58:26 UTC