

Name : Vansh Mittal
Contact No. : +91 9927220660
Email : vanshmittal1701@gmail.com
LinkedIn : <https://www.linkedin.com/in/vansh-mittal-51432b224/>
GitHub : [vansh1701 \(github.com\)](https://github.com/vansh1701)

Global Trend Programming Profile Assessment Questions

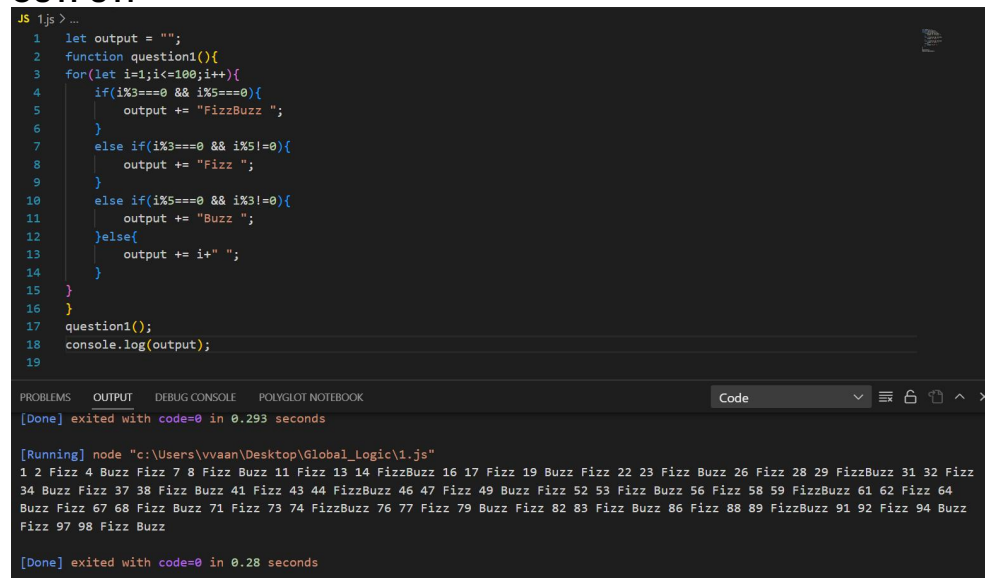
Questions

1. Write a function that prints the numbers from 1 to 100. But for multiples of three, print "Fizz" instead of the number, and for the multiples of five, print "Buzz". For numbers that are multiples of both three and five, print "FizzBuzz".

CODE:

```
let output = "";
function question1(){
  for(let i=1;i<=100;i++){
    if(i%3===0 && i%5===0){
      output += "FizzBuzz ";
    }
    else if(i%3===0 && i%5!=0){
      output += "Fizz ";
    }
    else if(i%5===0 && i%3!=0){
      output += "Buzz ";
    }else{
      output += i+" ";
    }
  }
}
question1();
console.log(output);
```

OUTPUT:



The screenshot shows a code editor with a dark theme. The code is written in JavaScript and is the same as the one provided in the 'CODE' section. Below the code editor, there is a 'PROBLEMS' tab and an 'OUTPUT' tab. The 'OUTPUT' tab is active, showing the result of running the code. The output is a single line of text: '1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz Fizz 22 23 Fizz Buzz 26 Fizz 28 29 FizzBuzz 31 32 Fizz 34 Buzz Fizz 37 38 Fizz Buzz 41 Fizz 43 44 FizzBuzz 46 47 Fizz 49 Buzz Fizz 52 53 Fizz Buzz 56 Fizz 58 59 FizzBuzz 61 62 Fizz 64 Buzz Fizz 67 68 Fizz Buzz 71 Fizz 73 74 FizzBuzz 76 77 Fizz 79 Buzz Fizz 82 83 Fizz Buzz 86 Fizz 88 89 FizzBuzz 91 92 Fizz 94 Buzz Fizz 97 98 Fizz Buzz'. The output is displayed in a monospace font, with each number or string separated by a space. The 'OUTPUT' tab also shows the execution time: '[Done] exited with code=0 in 0.293 seconds'.

2. Write a function that takes a string input representing a simple arithmetic expression (only addition and subtraction) and returns the result.

CODE :

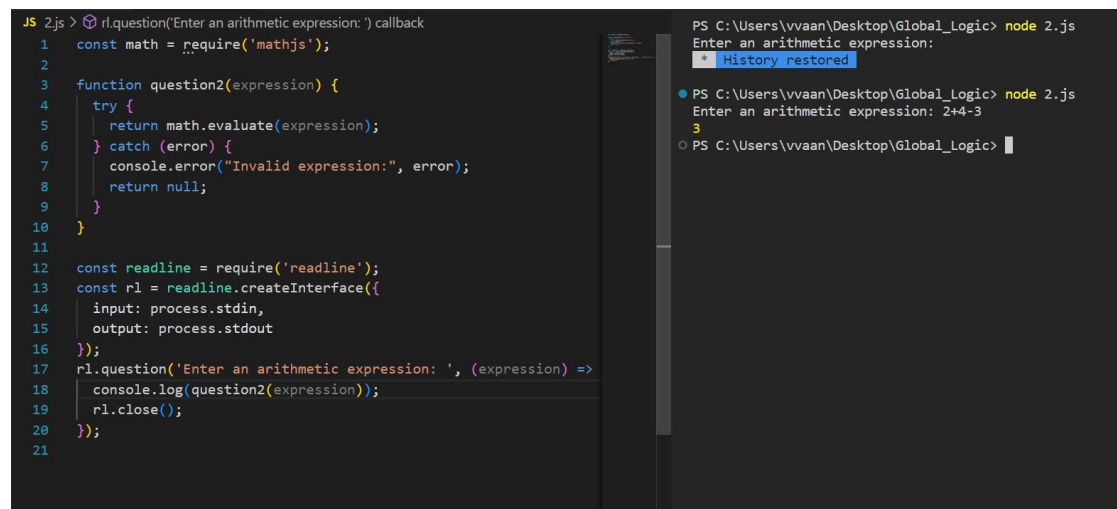
```
const math = require('mathjs');

function question2(expression) {
  try {
    return math.evaluate(expression);
  } catch (error) {
    console.error("Invalid expression:", error);
    return null;
  }
}

const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

rl.question('Enter an arithmetic expression: ', (expression) => {
  console.log(question2(expression));
  rl.close();
});
```

OUTPUT :



The screenshot shows a code editor on the left with the JavaScript code for the arithmetic evaluator. The code defines a `question2` function that uses `mathjs` to evaluate expressions, and a `readline` interface to prompt the user. The code is saved as `2.js`. On the right, a terminal window shows the command `node 2.js` being executed. The prompt 'Enter an arithmetic expression:' is displayed, and the user has entered '2+4-3'. The output shows the result '3'.

3. Write a function that takes a nested array and returns a flattened array.

CODE :

```
function question3(arr) {
  let tmp = [];
  function flatten(element) {
    if (Array.isArray(element)) {
      for (let item of element) {
        flatten(item);
      }
    } else {
```

```

        tmp.push(element);
    }
}
flatten(arr);
return tmp;
}
const arr = [1, [2, [3, 4], 5], [6, 7], 8];
console.log(question3(arr));

```

OUTPUT :

```

JS 3.js > ...
1  function question3(arr) {
2      let tmp = [];
3      function flatten(element) {
4          if (Array.isArray(element)) {
5              for (let item of element) {
6                  flatten(item);
7              }
8          } else {
9              tmp.push(element);
10         }
11     }
12     flatten(arr);
13     return tmp;
14 }
15 const arr = [1, [2, [3, 4], 5], [6, 7], 8];
16 console.log(question3(arr));
17

```

PROBLEMS OUTPUT DEBUG CONSOLE POLYGLOT NOTEBOOK: VARIABLES Code

[Done] exited with code=0 in 0.14 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\3.js"

```

[
  1, 2, 3, 4,
  5, 6, 7, 8
]

```

[Done] exited with code=0 in 0.197 seconds

4. Write a function that checks if two given strings are anagrams of each other.

CODE :

```

function areAnagrams(str1, str2) {
    return str1.split('').sort().join('') ===
    str2.split('').sort().join(''); }
let str1 = "care"
let str2 = "race"
console.log(areAnagrams(str1, str2));

```

OUTPUT :

```
JS 4js > ...
1 function question4(str1, str2) {
2   return str1.split('').sort().join('') === str2.split('').sort().join('');
3 }
4 let str1 = "care"
5 let str2 = "race"
6 console.log(question4(str1, str2));
```

PROBLEMS OUTPUT DEBUG CONSOLE POLYGLOT NOTEBOOK: VARIABLES Code

[Done] exited with code=0 in 0.287 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\3.js"
true

[Done] exited with code=0 in 0.332 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\3.js"
true

[Done] exited with code=0 in 0.137 seconds

5. Write a function that takes an array and returns a new array with duplicates removed.

CODE :

```
function removeDuplicates(arr) {
  return [...new Set(arr)];
}
let arr = [1,2,4,1,3,4,5,6];
console.log(removeDuplicates(arr));
```

OUTPUT :

```
JS 5js > [0] arr
1 function removeDuplicates(arr) {
2   return [...new Set(arr)];
3 }
4 let arr = [1,2,4,1,3,4,5,6];
5 console.log(removeDuplicates(arr));
```

PROBLEMS OUTPUT DEBUG CONSOLE POLYGLOT NOTEBOOK: VARIABLES Code

[Done] exited with code=0 in 0.194 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\5.js"
[1, 2, 3, 4, 5, 6]

[Done] exited with code=0 in 0.511 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\5.js"
[1, 2, 4, 3, 5, 6]

[Done] exited with code=0 in 0.306 seconds

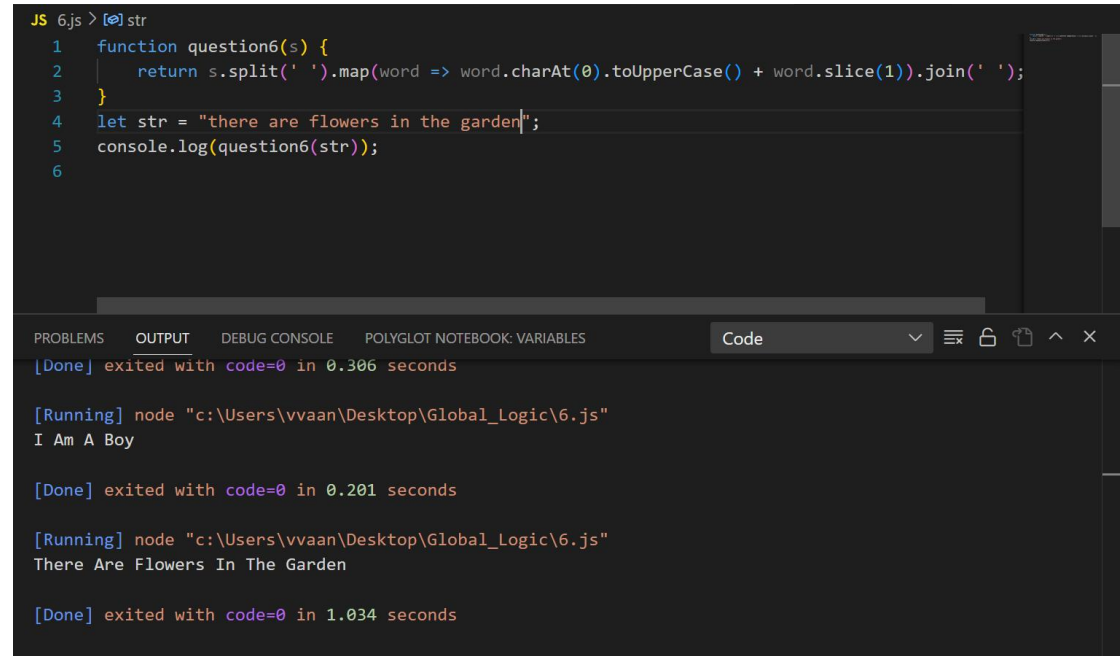
6. Write a function that takes a string and capitalizes the first letter of each word in the string.

CODE :

```
function question6(s) {
```

```
        return s.split(' ').map(word => word.charAt(0).toUpperCase() +
word.slice(1)).join(' ');
    }
    let str = "there are flowers in the garden";
    console.log(question6(str));
```

OUTPUT :



The screenshot shows a code editor with a dark theme. The code for `question6` is written in JavaScript. Below the code, the output is displayed in the console. The console shows three separate runs of the code, each with its own output and execution time.

```
JS 6.js > [?] str
1 function question6(s) {
2   |   return s.split(' ').map(word => word.charAt(0).toUpperCase() + word.slice(1)).join(' ');
3   | }
4 let str = "there are flowers in the garden";
5 console.log(question6(str));
6

[Done] exited with code=0 in 0.306 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\6.js"
I Am A Boy

[Done] exited with code=0 in 0.201 seconds

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\6.js"
There Are Flowers In The Garden

[Done] exited with code=0 in 1.034 seconds
```

7. Write a function that generates the first n numbers of the Fibonacci sequence.

CODE :

```
function question7(n) {
    let tmp = [];
    let a = 0, b = 1;
    for (let i = 0; i < n; i++) {
        tmp.push(a);
        [a, b] = [b, a + b];
    }
    return tmp;
}
console.log(question7(10));
```

OUTPUT :

```
JS 7.js > question7
1 function question7(n) {
2   let tmp = [];
3   let a = 0, b = 1;
4   for (let i = 0; i < n; i++) {
5     tmp.push(a);
6     [a, b] = [b, a + b];
7   }
8   return tmp;
9 }
10 console.log(question7(10));
11

PROBLEMS OUTPUT DEBUG CONSOLE POLYGLOT NOTEBOOK: VARIABLES Code
[Done] exited with code=0 in 0.189 seconds
[Running] node "c:\Users\vvaan\Desktop\Global_Logic\7.js"
[
  0, 1, 1, 2, 3,
  5, 8, 13, 21, 34
]
[Done] exited with code=0 in 0.149 seconds
```

8. Implement a simple HashMap class with *put*, *get*, and *remove* methods.

CODE :

```
class HashMap {
  constructor() {
    this.map = {};
  }
  put(key, value) {
    const hash = key.toString().length % this.map.length;
    if (!this.map[hash]) {
      this.map[hash] = [];
    }
    this.map[hash].push([key, value]);
  }
  get(key) {
    const hash = key.toString().length % this.map.length;
    const tmp = this.map[hash];
    if (tmp) {
      for (const [k, v] of tmp) {
        if (k === key) {
          return v;
        }
      }
    }
    return null;
  }
  remove(key) {
    const hash = key.toString().length % this.map.length;
    const tmp = this.map[hash];
    if (tmp) {
      for (let i = 0; i < tmp.length; i++) {
        if (tmp[i][0] === key) {
          tmp.splice(i, 1);
          return true;
        }
      }
    }
  }
}
```

```

    }
  }
  return false;
}
}
const question8 = new HashMap();
const items = [
  ["name", "Vansh Mittal"],
  ["age", 21],
  ["city", "Dehradun"],
];
for (const item of items) {
  const [key, value] = item;
  question8.put(key, value);
}
const name = question8.get("name");
console.log("Name:", name);
const removedAge = question8.remove("age");
if (removedAge) {
  console.log("Age removed successfully.");
} else {
  console.log("Age not found in the map.");
}

```

OUTPUT :



The screenshot shows a VS Code editor with a file named `8.js`. The code defines a `HashMap` class with methods `put`, `get`, and `remove`. The `put` method uses a hash-based approach to store key-value pairs. The `get` method retrieves the value for a given key. The `remove` method removes a key-value pair from the map. The output console shows the following results:

```

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\8.js"
Name: Vansh Mittal
Age removed successfully.
[Done] exited with code=0 in 0.19 seconds

```

9. Write a function that filters out even numbers from an array.

CODE :

```

function question9(arr) {
  return arr.filter(num => num % 2 !== 0);
}
let arr = [1,2,3,4,5,6,7,8,9,10]
console.log(question9(arr));

```

OUTPUT :

```
JS 9.js > ...
1 function question9(arr) {
2   return arr.filter(num => num % 2 !== 0);
3 }
4 let arr = [1,2,34,5,6,7,8,9,10]
5 console.log(question9(arr));
```

PROBLEMS OUTPUT DEBUG CONSOLE POLYGLOT NOTEBOOK: VARIABLES Code

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\9.js"

[1, 5, 7, 9]

[Done] exited with code=0 in 0.228 seconds

10. Write a function that converts a given string to title case (capitalizing the first letter of each word).

CODE :

```
function question10(s) {
  return s.split(' ').map(word => word.charAt(0).toUpperCase() +
word.slice(1)).join(' ');
}
let s = "i want to get settle in new york."
console.log(question10(s));
```

OUTPUT :

```
JS 10.js > ...
1 function question10(s) {
2   return s.split(' ').map(word => word.charAt(0).toUpperCase() + word.slice(1)).join(' ');
3 }
4 let s = "i want to get settle in new york."
5 console.log(question10(s));
```

PROBLEMS OUTPUT DEBUG CONSOLE POLYGLOT NOTEBOOK: VARIABLES Code

[Running] node "c:\Users\vvaan\Desktop\Global_Logic\10.js"

I Want To Get Settle In New York.

[Done] exited with code=0 in 0.33 seconds