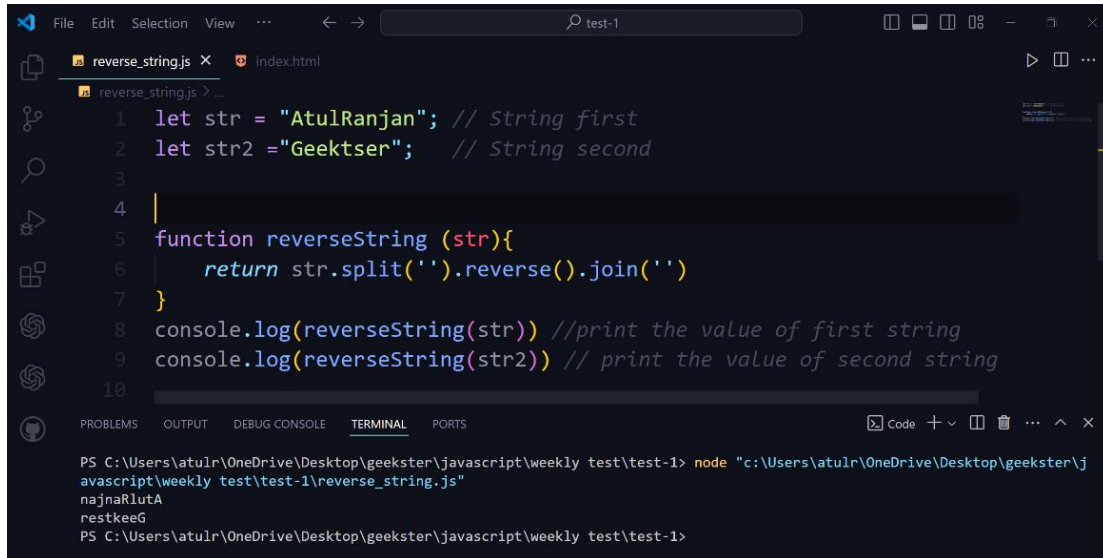


Name – Atul Ranjan

Weekly test 1(fs-18)

1 String Reversal



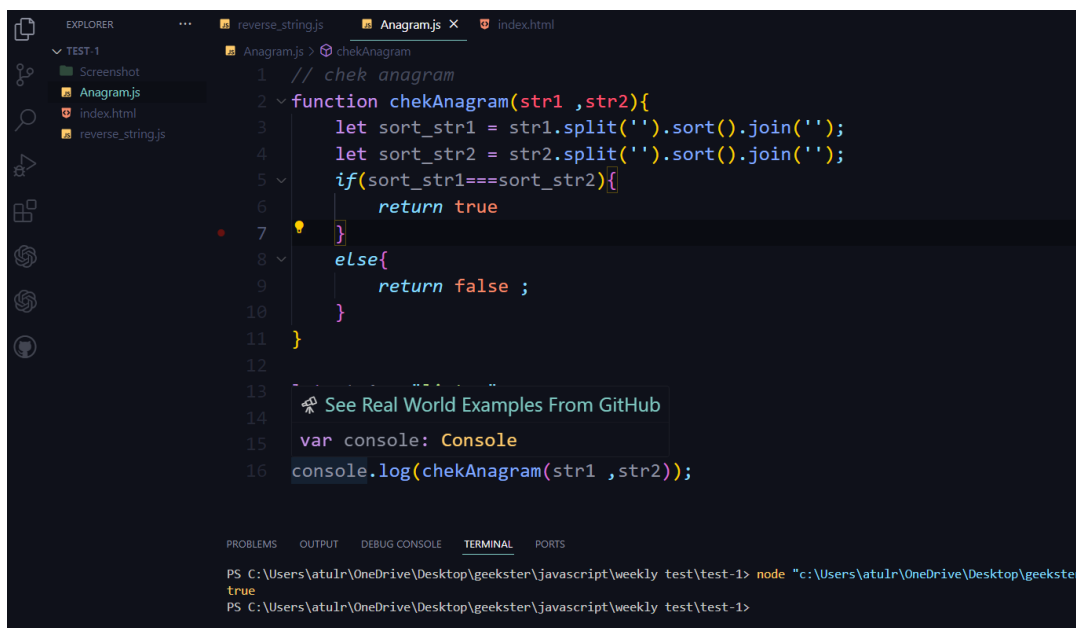
```
1 let str = "AtulRanjan"; // String first
2 let str2 = "Geektser"; // String second
3
4
5 function reverseString (str){
6     return str.split('').reverse().join('')
7 }
8 console.log(reverseString(str)) //print the value of first string
9 console.log(reverseString(str2)) // print the value of second string
10
```

PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1> node "c:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1\reverse_string.js"

najnaRluta
restkeeG

PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1>

2 Anagram Check



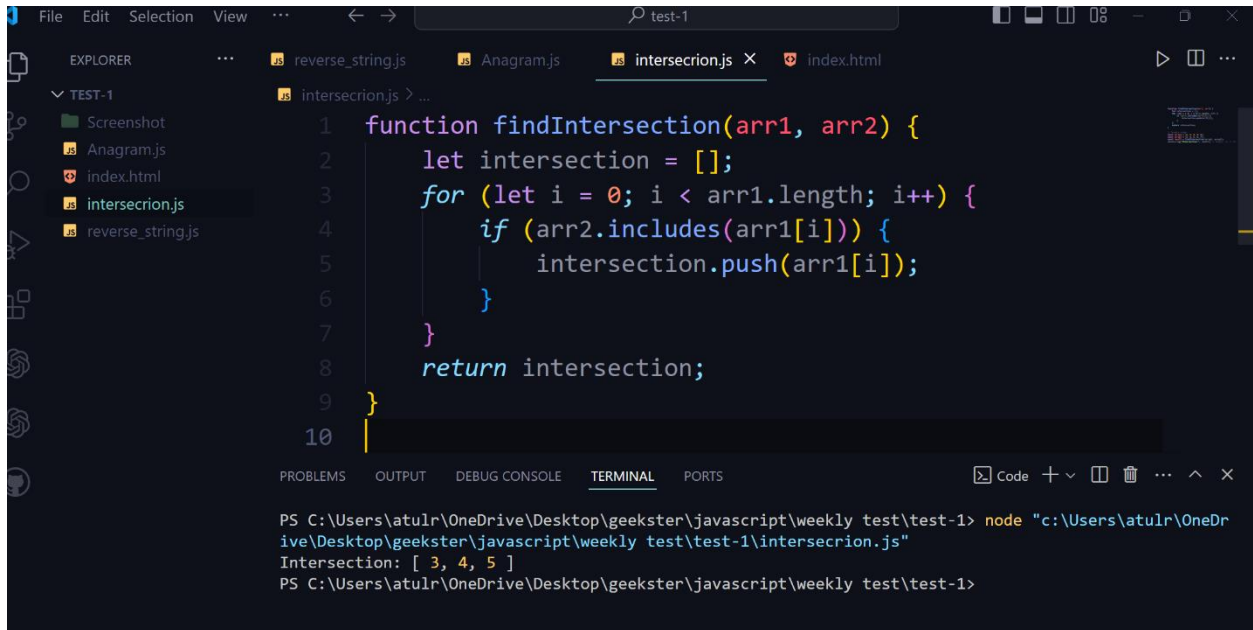
```
1 // chek anagram
2 function chekAnagram(str1 ,str2){
3     let sort_str1 = str1.split('').sort().join('');
4     let sort_str2 = str2.split('').sort().join('');
5     if(sort_str1===sort_str2){
6         return true
7     }
8     else{
9         return false ;
10    }
11 }
12
13
14
15
16 console.log(chekAnagram(str1 ,str2));
```

PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1> node "c:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1\Anagram.js"

true

PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1>

3 Array Intersection



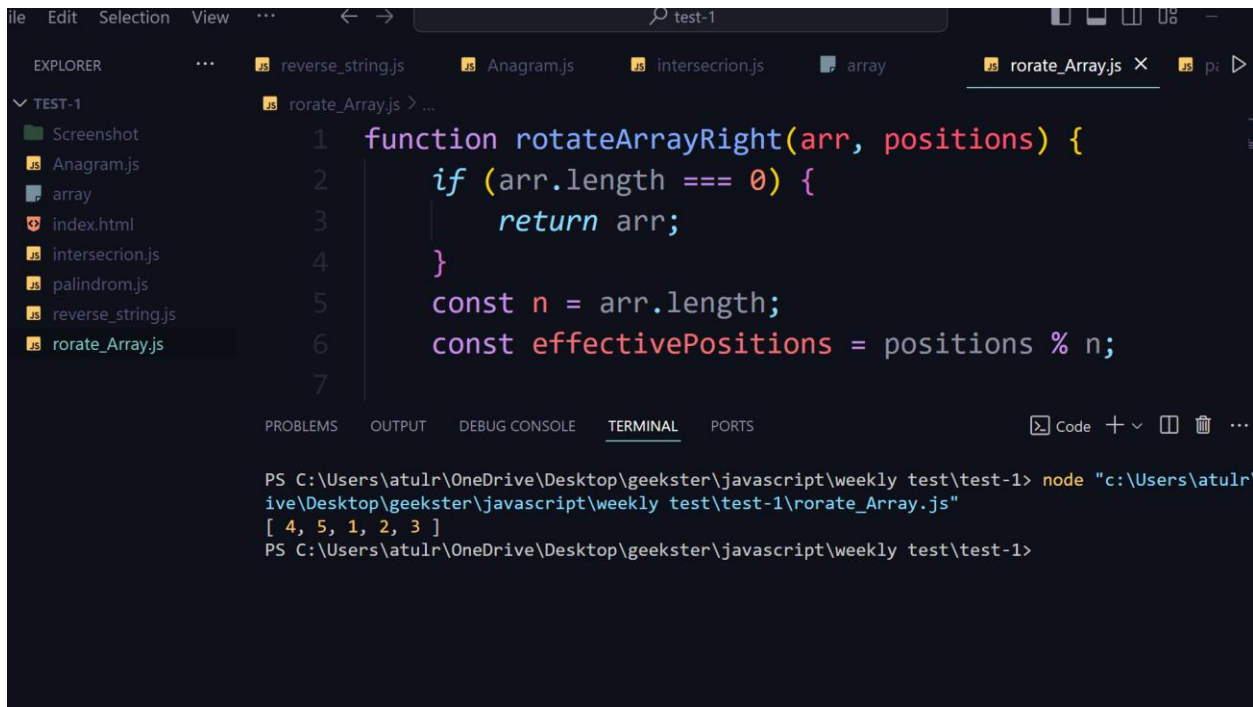
The screenshot shows a VS Code editor with a file explorer on the left containing files like 'Screenshot', 'Anagram.js', 'index.html', 'intersection.js', and 'reverse_string.js'. The main editor displays the code for 'intersection.js'.

```
1 function findIntersection(arr1, arr2) {  
2   let intersection = [];  
3   for (let i = 0; i < arr1.length; i++) {  
4     if (arr2.includes(arr1[i])) {  
5       intersection.push(arr1[i]);  
6     }  
7   }  
8   return intersection;  
9 }  
10
```

The terminal at the bottom shows the command to run the script and its output:

```
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1> node "c:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1\intersection.js"  
Intersection: [ 3, 4, 5 ]  
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1>
```

4 Array Rotation



The screenshot shows a VS Code editor with a file explorer on the left containing files like 'Screenshot', 'Anagram.js', 'array', 'index.html', 'intersection.js', 'palindrom.js', 'reverse_string.js', and 'rorate_Array.js'. The main editor displays the code for 'rorate_Array.js'.

```
1 function rotateArrayRight(arr, positions) {  
2   if (arr.length === 0) {  
3     return arr;  
4   }  
5   const n = arr.length;  
6   const effectivePositions = positions % n;  
7
```

The terminal at the bottom shows the command to run the script and its output:

```
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1> node "c:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1\rorate_Array.js"  
[ 4, 5, 1, 2, 3 ]  
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1>
```

5 String palindrome

The image shows a VS Code editor window with a file named 'palindrom.js' open. The file contains a JavaScript function 'isPalindrome' that checks if a string is a palindrome. The function uses two pointers, 'start' and 'end', to compare characters from both ends of the string. If any pair of characters does not match, it returns 'false'; otherwise, it returns 'true'. Below the function, there are three console.log statements demonstrating its usage with the strings 'A man, a plan, a canal, Panama', 'race car', and 'hello'. The terminal at the bottom shows the command 'node "c:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1\palindrom.js"' being executed, resulting in the output: 'false', 'true', and 'false' on separate lines. The VS Code interface includes a sidebar with file explorer, a top bar with file names, and a bottom bar with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'.

6 String compression

```
1 function compressString(str) {
2     let compressed = '';
3     let count = 1;
4     for (let i = 0; i < str.length; i++) {
5         if (str[i] === str[i + 1]) {
6             count++;
7         } else {
8             compressed += str[i] + count;
9             count = 1;
10        }
11    }
12    return compressed.length < str.length ? compressed : str;
13 }
14 console.log(compressString("aabccccaaa")); // Output: "a2b1c5a3"
```

7 Array sum

```
targetSum.js > ...
1 function Targetsum(arr, target) {
2   let i = 0;
3   let j = arr.length - 1;
4
5   // Sorting the array in ascending order
6   arr.sort((a, b) => a - b);
7
8   while (i < j) {
9     let sum = arr[i] + arr[j];
10    if (sum === target) {
11      console.log(arr[i], arr[j]); // Print the pair that sums up to the target
12      i++; // Move i to the right to find the next pair
13      j--; // Move j to the left to find the next pair
14    } else if (sum < target) {
15      i++; // If the sum is smaller than the target, move i to the right to increase the sum
16    } else {
17      j--; // If the sum is greater than the target, move j to the left to decrease the sum
18    }
19  }
20 }
21
22 // Example usage:
23 Targetsum([1,2,3,4,5,6,7,8], 9);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code +

```
1 8
2 7
3 6
4 5
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1>
```

8 Longest substring without repeating character

```
longest_Substring.js > ...
1 function longestSubstringWithoutRepeatingCharacters(str) {
2   let maxLength = 0;
3   let start = 0;
4   const charMap = {};
5
6   for (let end = 0; end < str.length; end++) {
7     const currentChar = str[end];
8     if (charMap[currentChar] !== undefined && charMap[currentChar] >= start) {
9       start = charMap[currentChar] + 1;
10    }
11    charMap[currentChar] = end;
12    maxLength = Math.max(maxLength, end - start + 1);
13  }
14
15  return maxLength;
16 }
17
18 // Example usage:
19 console.log(longestSubstringWithoutRepeatingCharacters("abcabcbb")); // Output: 3
20 console.log(longestSubstringWithoutRepeatingCharacters("bbbbbb")); // Output: 1
21 console.log(longestSubstringWithoutRepeatingCharacters("pwwkew")); // Output: 3
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code +

```
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1> node "c:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1\longest_Substring.js"
3
3
PS C:\Users\atulr\OneDrive\Desktop\geekster\javascript\weekly test\test-1>
```