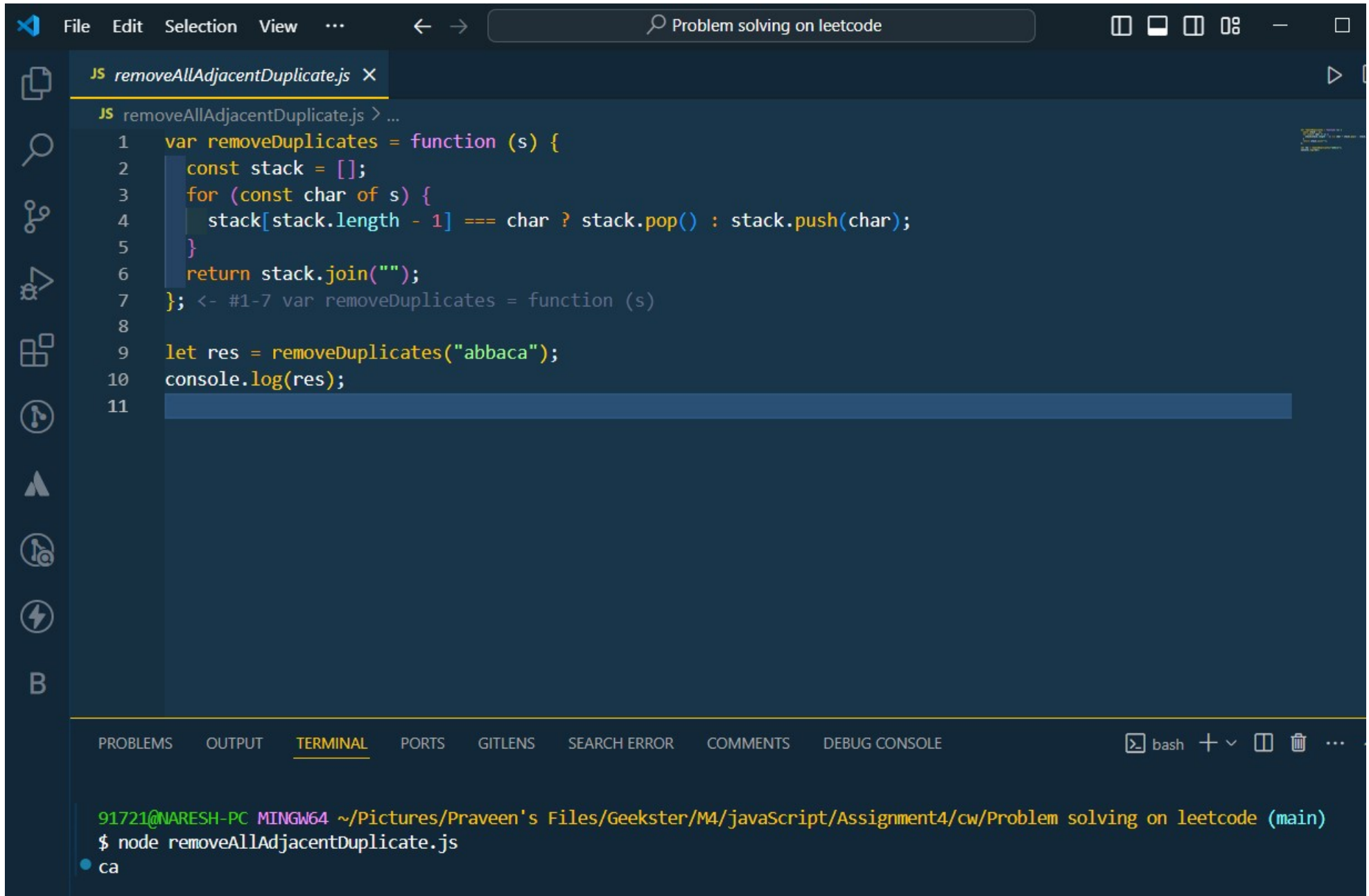


Note : "All the solutions to the questions are provided, and we also have the output of these questions in the terminal."

1.) Remove all adjacent duplicates from a string :

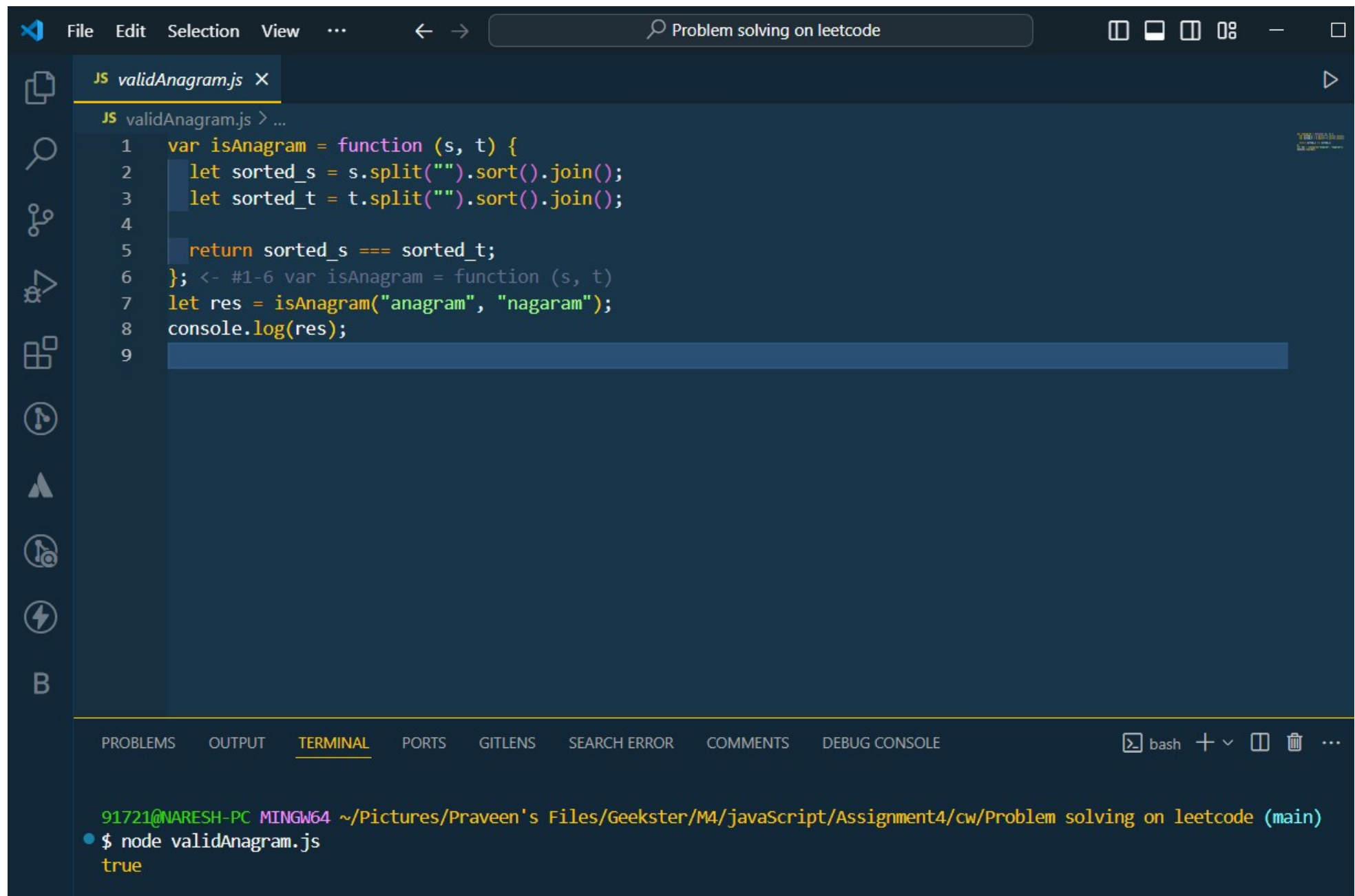


```
JS removeAllAdjacentDuplicate.js X
JS removeAllAdjacentDuplicate.js > ...
1  var removeDuplicates = function (s) {
2      const stack = [];
3      for (const char of s) {
4          stack[stack.length - 1] === char ? stack.pop() : stack.push(char);
5      }
6      return stack.join("");
7  }; <- #1-7 var removeDuplicates = function (s)
8
9  let res = removeDuplicates("abbaca");
10 console.log(res);
11
```

PROBLEMS OUTPUT TERMINAL PORTS GITLENS SEARCH ERROR COMMENTS DEBUG CONSOLE

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javascript/Assignment4/cw/Problem solving on leetcode (main)
$ node removeAllAdjacentDuplicate.js
ca
```

2.) Valid Anagram :



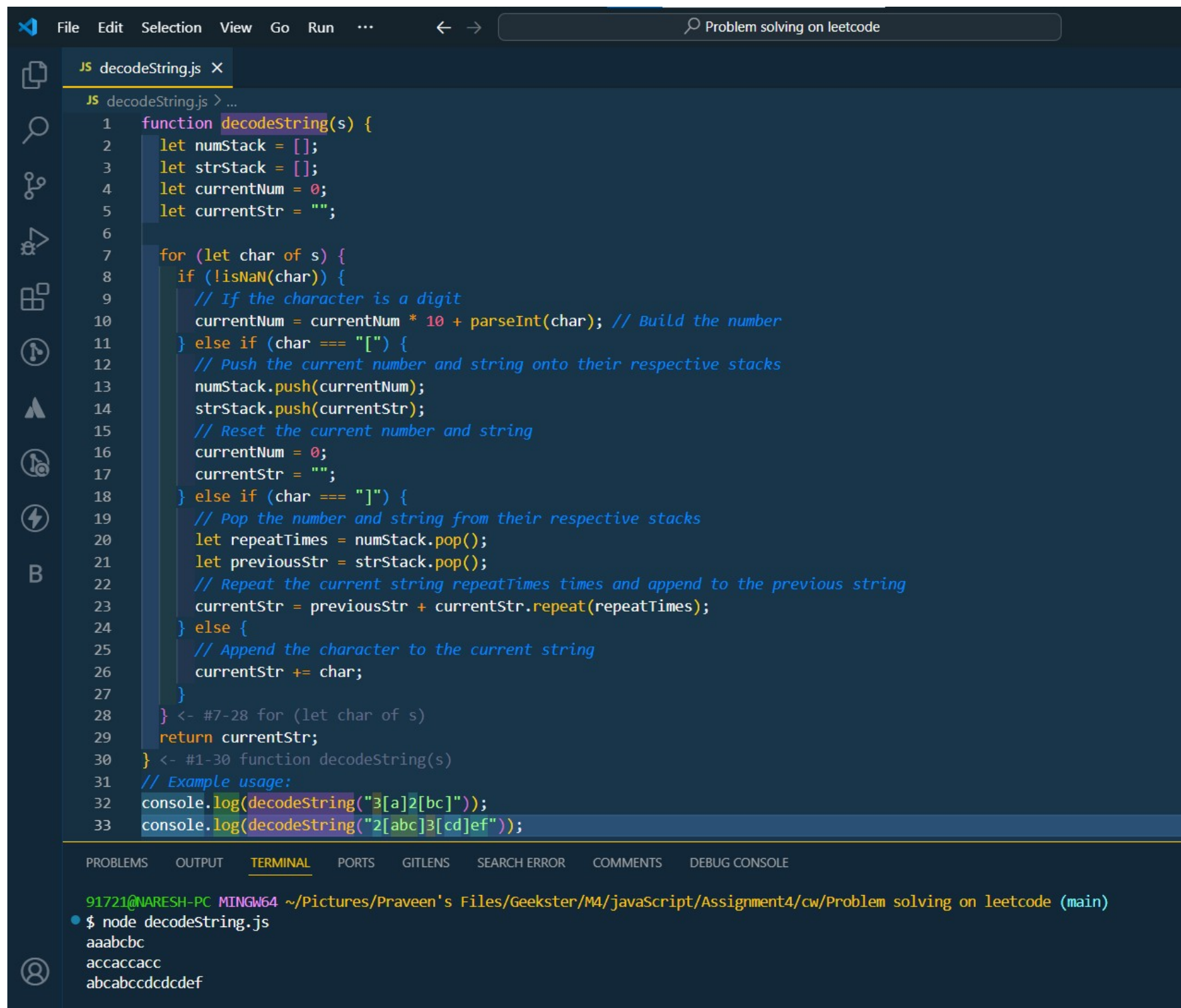
The image shows a Visual Studio Code editor window with a dark theme. The top menu bar includes File, Edit, Selection, View, and a search bar containing "Problem solving on leetcode". The editor is open to a file named `validAnagram.js`. The code in the file is as follows:

```
1 var isAnagram = function (s, t) {  
2   let sorted_s = s.split("").sort().join();  
3   let sorted_t = t.split("").sort().join();  
4  
5   return sorted_s === sorted_t;  
6 }; <- #1-6 var isAnagram = function (s, t)  
7 let res = isAnagram("anagram", "nagaram");  
8 console.log(res);  
9
```

Below the editor, the TERMINAL tab is active, showing the command prompt and the output of the script:

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javascript/Assignment4/cw/Problem solving on leetcode (main)  
• $ node validAnagram.js  
true
```

3.) Decode String :



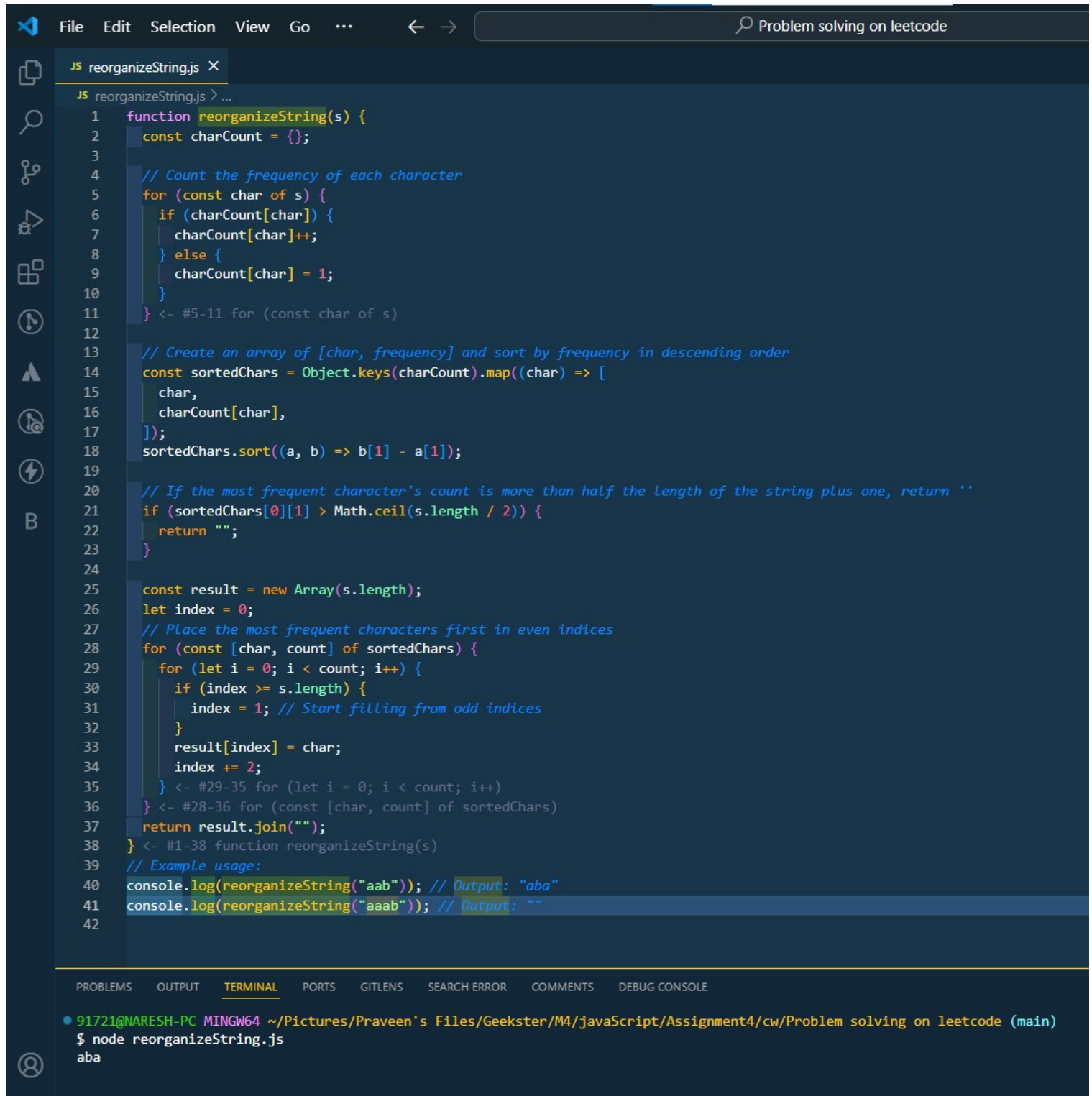
The image shows a VS Code editor window with a JavaScript file named `decodeString.js`. The code implements a function `decodeString(s)` that decodes a string with nested brackets and repeat counts. It uses two stacks, `numStack` and `strStack`, to manage the state. The function iterates through each character of the string. If it's a digit, it builds the repeat count. If it's an opening bracket `[`, it pushes the current number and string onto the stacks. If it's a closing bracket `]`, it pops the repeat count and previous string, then repeats the current string and appends it to the previous string. If it's any other character, it simply appends it to the current string. The function returns the final decoded string.

```
1 function decodeString(s) {
2   let numStack = [];
3   let strStack = [];
4   let currentNum = 0;
5   let currentStr = "";
6
7   for (let char of s) {
8     if (!isNaN(char)) {
9       // If the character is a digit
10      currentNum = currentNum * 10 + parseInt(char); // Build the number
11    } else if (char === "[") {
12      // Push the current number and string onto their respective stacks
13      numStack.push(currentNum);
14      strStack.push(currentStr);
15      // Reset the current number and string
16      currentNum = 0;
17      currentStr = "";
18    } else if (char === "]") {
19      // Pop the number and string from their respective stacks
20      let repeatTimes = numStack.pop();
21      let previousStr = strStack.pop();
22      // Repeat the current string repeatTimes times and append to the previous string
23      currentStr = previousStr + currentStr.repeat(repeatTimes);
24    } else {
25      // Append the character to the current string
26      currentStr += char;
27    }
28  } <- #7-28 for (let char of s)
29  return currentStr;
30 } <- #1-30 function decodeString(s)
31 // Example usage:
32 console.log(decodeString("3[a]2[bc]"));
33 console.log(decodeString("2[abc]3[cd]ef"));
```

The terminal output shows the execution of the code:

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javascript/Assignment4/cw/Problem solving on leetcode (main)
$ node decodeString.js
aaabcbc
accaccacc
abcabccdcdcdef
```

4.) Reorganize String :



```
File Edit Selection View Go ... Problem solving on leetcode

JS reorganizeString.js X
JS reorganizeString.js > ...
1 function reorganizeString(s) {
2   const charCount = {};
3
4   // Count the frequency of each character
5   for (const char of s) {
6     if (charCount[char]) {
7       charCount[char]++;
8     } else {
9       charCount[char] = 1;
10    }
11  } <- #5-11 for (const char of s)
12
13  // Create an array of [char, frequency] and sort by frequency in descending order
14  const sortedChars = Object.keys(charCount).map((char) => [
15    char,
16    charCount[char],
17  ]);
18  sortedChars.sort((a, b) => b[1] - a[1]);
19
20  // If the most frequent character's count is more than half the length of the string plus one, return ''
21  if (sortedChars[0][1] > Math.ceil(s.length / 2)) {
22    return "";
23  }
24
25  const result = new Array(s.length);
26  let index = 0;
27  // Place the most frequent characters first in even indices
28  for (const [char, count] of sortedChars) {
29    for (let i = 0; i < count; i++) {
30      if (index >= s.length) {
31        index = 1; // Start filling from odd indices
32      }
33      result[index] = char;
34      index += 2;
35    } <- #29-35 for (let i = 0; i < count; i++)
36  } <- #28-36 for (const [char, count] of sortedChars)
37  return result.join("");
38 } <- #1-38 function reorganizeString(s)
39 // Example usage:
40 console.log(reorganizeString("aab")); // Output: "aba"
41 console.log(reorganizeString("aaab")); // Output: ""
42
```

PROBLEMS OUTPUT **TERMINAL** PORTS GITLENS SEARCH ERROR COMMENTS DEBUG CONSOLE

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javascript/Assignment4/cw/Problem solving on leetcode (main)
$ node reorganizeString.js
aba
```