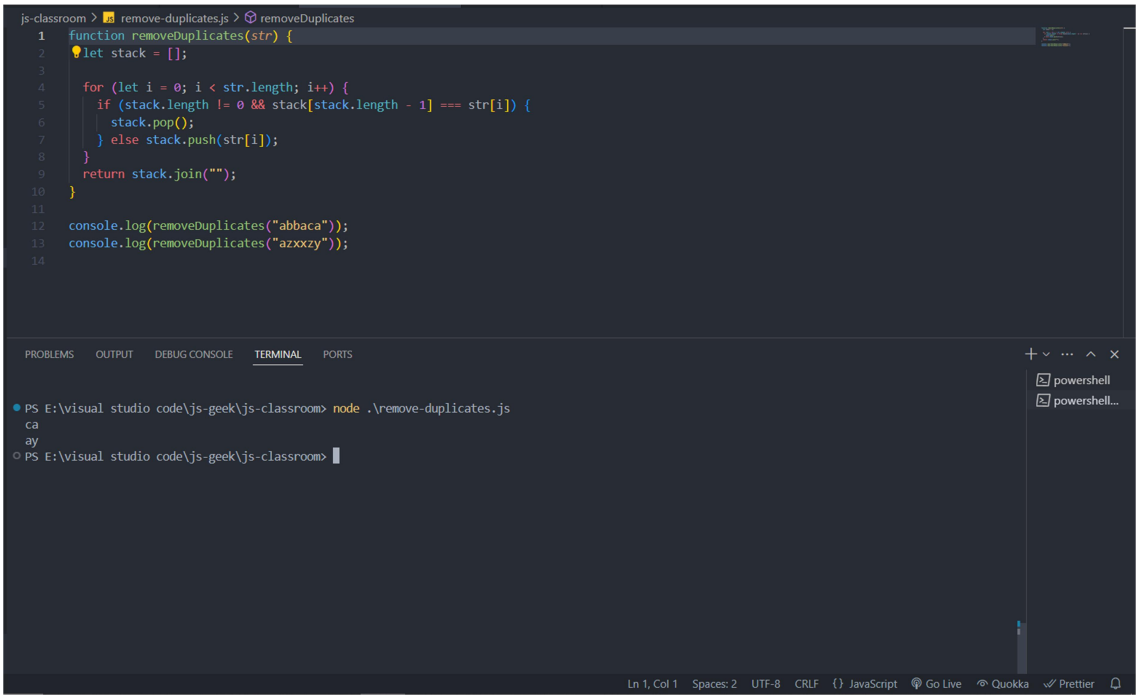
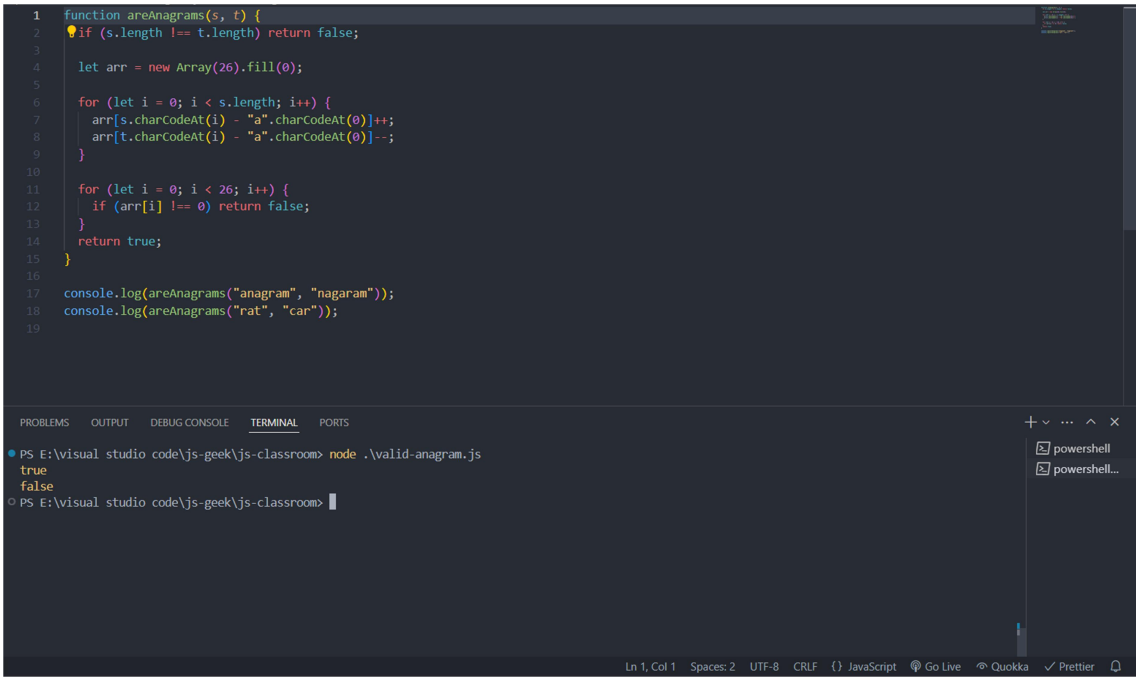
Ques1)



```
function removeDuplicates(str) {
  let stack = [];

  for (let i = 0; i < str.length; i++) {
    if (stack.length != 0 && stack[stack.length - 1] === str[i]) {
      stack.pop();
    } else stack.push(str[i]);
  }
  return stack.join("");
}

console.log(removeDuplicates("abbaca"));
console.log(removeDuplicates("azxxzy"));
```

```
PS E:\visual studio code\js-geek\js-classroom> node .\remove-duplicates.js
ca
ay
PS E:\visual studio code\js-geek\js-classroom>
```

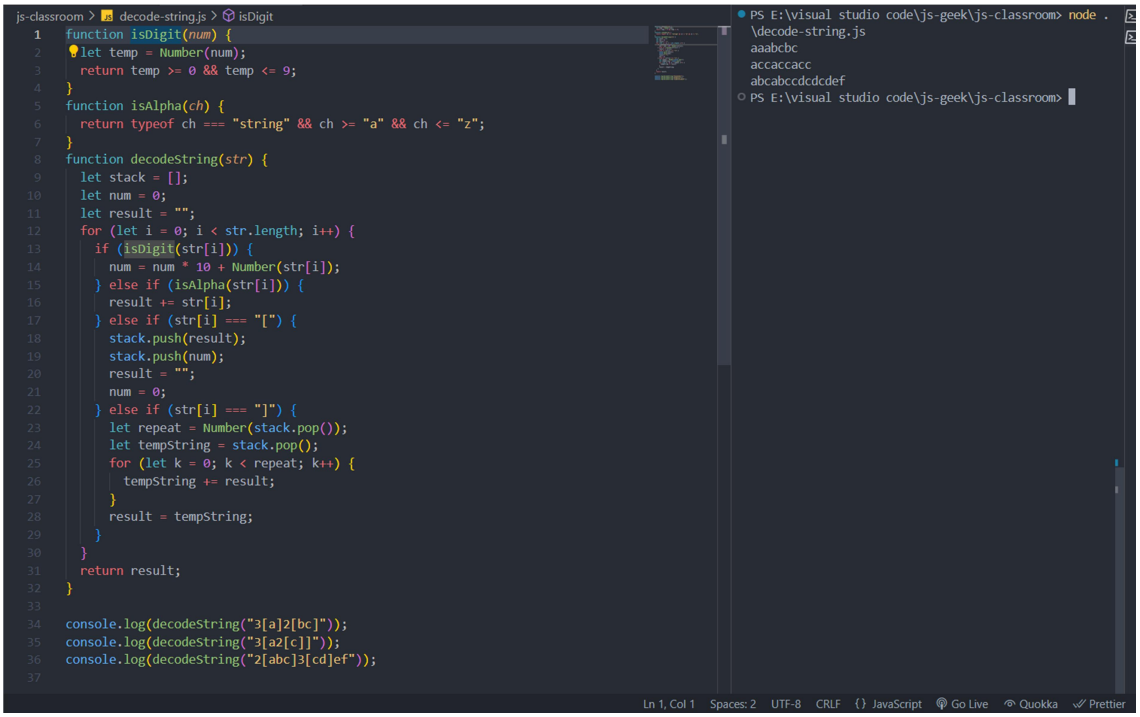
Ques2)



```
function areAnagrams(s, t) {
  if (s.length !== t.length) return false;

  let arr = new Array(26).fill(0);

  for (let i = 0; i < s.length; i++) {
    arr[s.charCodeAt(i) - "a".charCodeAt(0)]++;
    arr[t.charCodeAt(i) - "a".charCodeAt(0)]--;
  }

  for (let i = 0; i < 26; i++) {
    if (arr[i] !== 0) return false;
  }
  return true;
}

console.log(areAnagrams("anagram", "nagaram"));
console.log(areAnagrams("rat", "car"));
```

```
PS E:\visual studio code\js-geek\js-classroom> node .\valid-anagram.js
true
false
PS E:\visual studio code\js-geek\js-classroom>
```

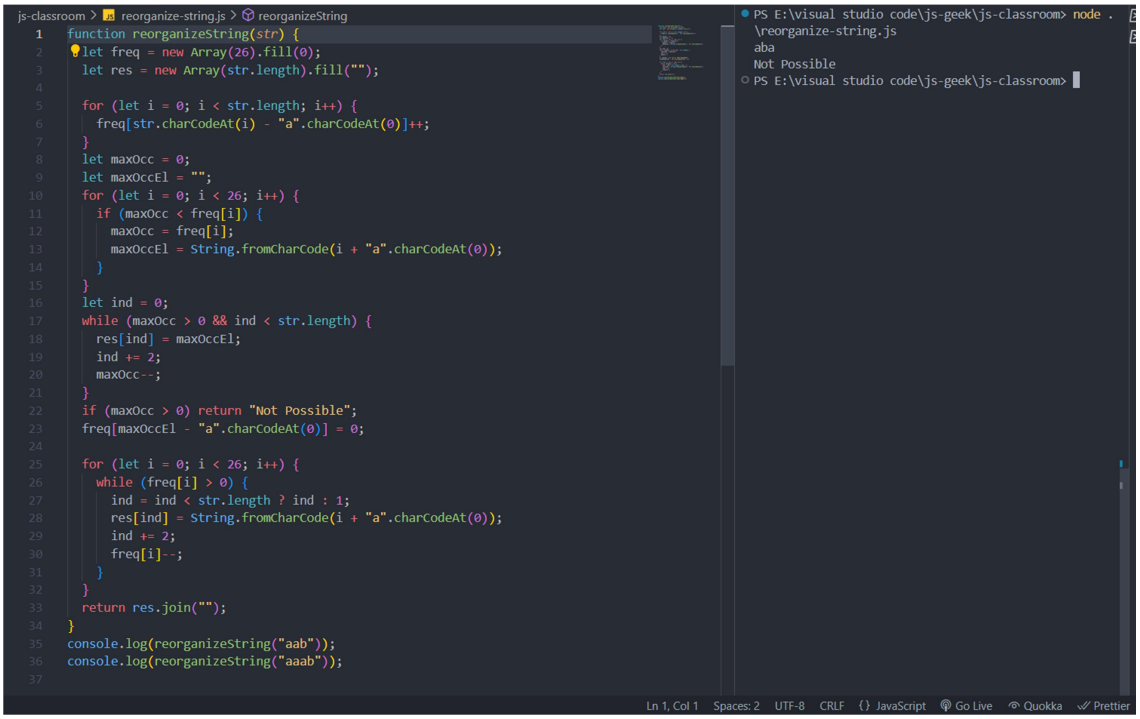
Ques3)



```
function isDigit(num) {
  let temp = Number(num);
  return temp >= 0 && temp <= 9;
}
function isAlpha(ch) {
  return typeof ch === "string" && ch >= "a" && ch <= "z";
}
function decodeString(str) {
  let stack = [];
  let num = 0;
  let result = "";
  for (let i = 0; i < str.length; i++) {
    if (isDigit(str[i])) {
      num = num * 10 + Number(str[i]);
    } else if (isAlpha(str[i])) {
      result += str[i];
    } else if (str[i] === "[") {
      stack.push(result);
      stack.push(num);
      result = "";
      num = 0;
    } else if (str[i] === "]") {
      let repeat = Number(stack.pop());
      let tempString = stack.pop();
      for (let k = 0; k < repeat; k++) {
        tempString += result;
      }
      result = tempString;
    }
  }
  return result;
}

console.log(decodeString("3[a]2[bc]"));
console.log(decodeString("3[a2[c]]"));
console.log(decodeString("2[abc]3[cd]ef"));
```

```
PS E:\visual studio code\js-geek\js-classroom> node .
\decode-string.js
aaabcbc
accaccacc
abcabccdcdcdef
PS E:\visual studio code\js-geek\js-classroom>
```

Ques4)



```
function reorganizeString(str) {
  let freq = new Array(26).fill(0);
  let res = new Array(str.length).fill("");

  for (let i = 0; i < str.length; i++) {
    freq[str.charCodeAt(i) - "a".charCodeAt(0)]++;
  }
  let maxOcc = 0;
  let maxOccEl = "";
  for (let i = 0; i < 26; i++) {
    if (maxOcc < freq[i]) {
      maxOcc = freq[i];
      maxOccEl = String.fromCharCode(i + "a".charCodeAt(0));
    }
  }
  let ind = 0;
  while (maxOcc > 0 && ind < str.length) {
    res[ind] = maxOccEl;
    ind += 2;
    maxOcc--;
  }
  if (maxOcc > 0) return "Not Possible";
  freq[maxOccEl - "a".charCodeAt(0)] = 0;

  for (let i = 0; i < 26; i++) {
    while (freq[i] > 0) {
      ind = ind < str.length ? ind : 1;
      res[ind] = String.fromCharCode(i + "a".charCodeAt(0));
      ind += 2;
      freq[i]--;
    }
  }
  return res.join("");
}
console.log(reorganizeString("aab"));
console.log(reorganizeString("aaab"));
```

```
PS E:\visual studio code\js-geek\js-classroom> node .
\reorganize-string.js
aba
Not Possible
PS E:\visual studio code\js-geek\js-classroom>
```