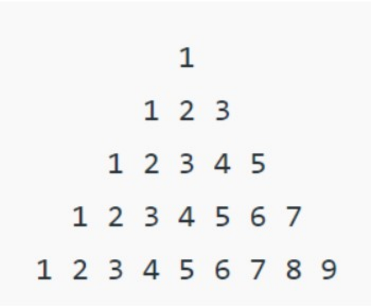
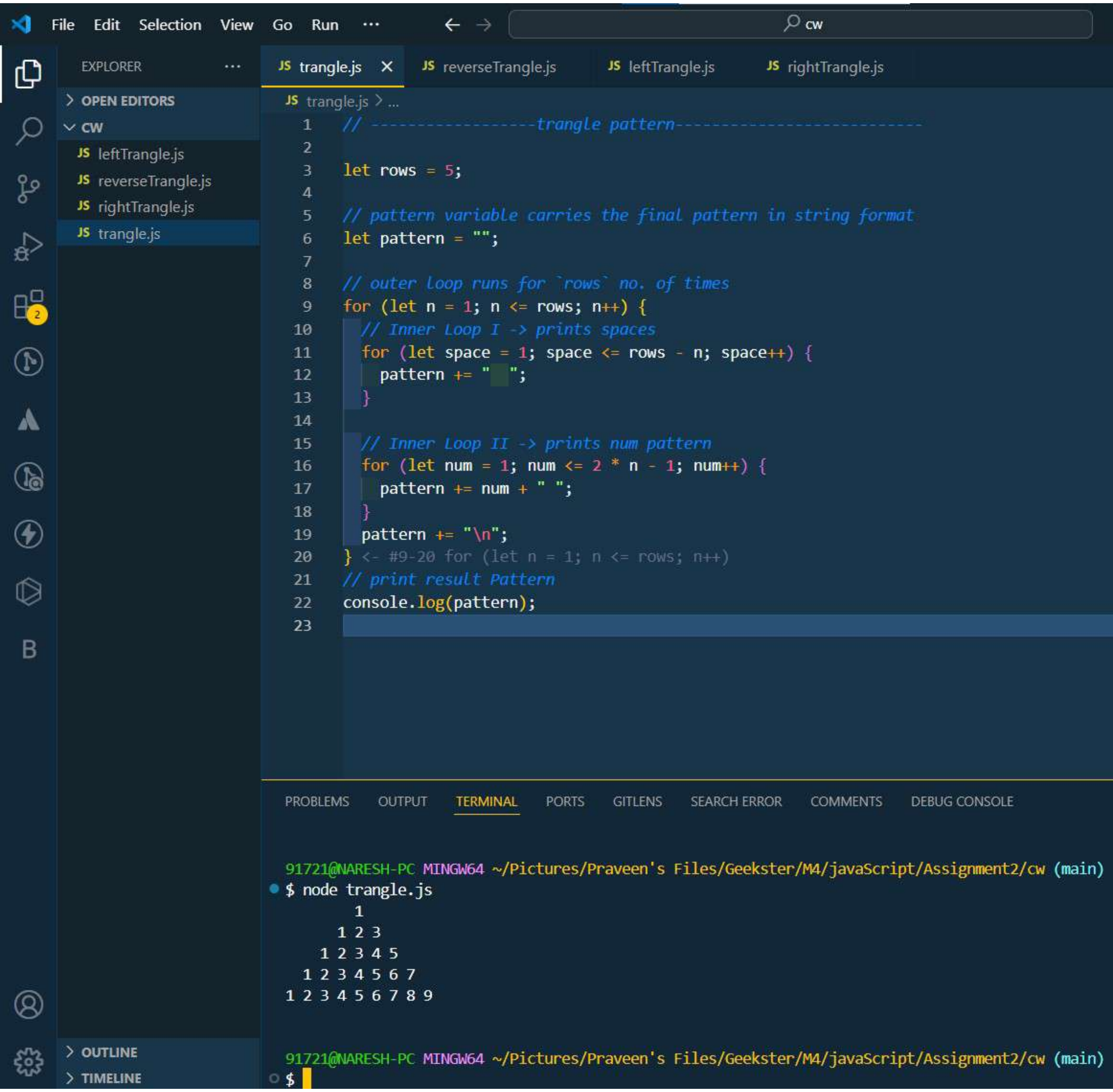
Q1.) Write a program in javascript to create a pattern like this



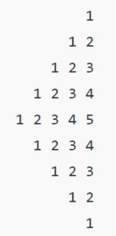
```
        1
      1 2 3
    1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9
```



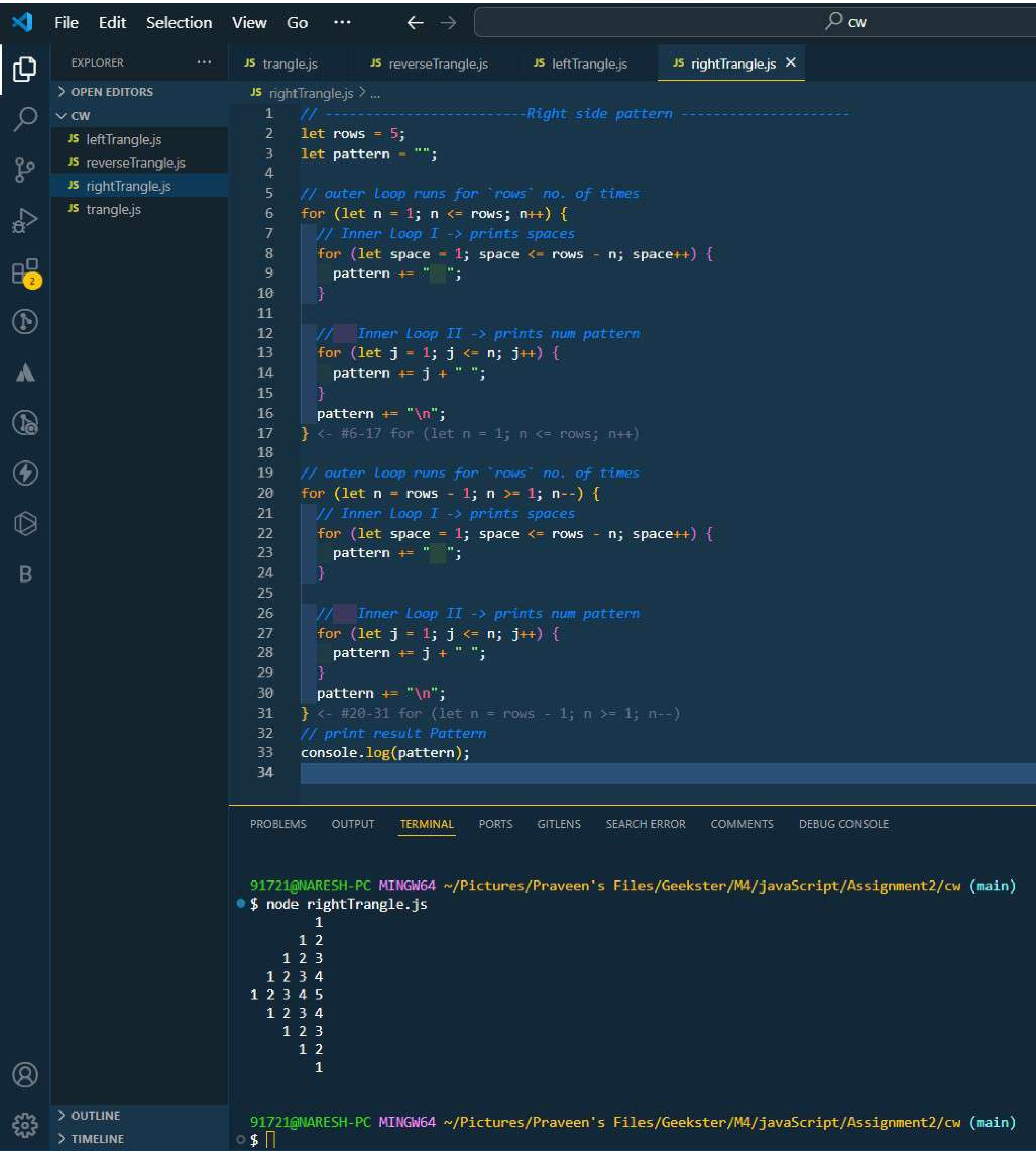
```javascript
// -----------------trangle pattern-------------------------

let rows = 5;

// pattern variable carries the final pattern in string format
let pattern = "";

// outer loop runs for `rows` no. of times
for (let n = 1; n <= rows; n++) {
  // Inner Loop I -> prints spaces
  for (let space = 1; space <= rows - n; space++) {
    pattern += "  ";
  }

  // Inner Loop II -> prints num pattern
  for (let num = 1; num <= 2 * n - 1; num++) {
    pattern += num + " ";
  }
  pattern += "\n";
} <- #9-20 for (let n = 1; n <= rows; n++)
// print result Pattern
console.log(pattern);
```

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ node trangle.js
        1
      1 2 3
    1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9


91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$
```

Q2.) Write a program in javascript to
create a pattern like this



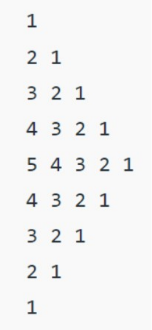
```
        1
      1 2
    1 2 3
  1 2 3 4
1 2 3 4 5
  1 2 3 4
    1 2 3
      1 2
        1
```



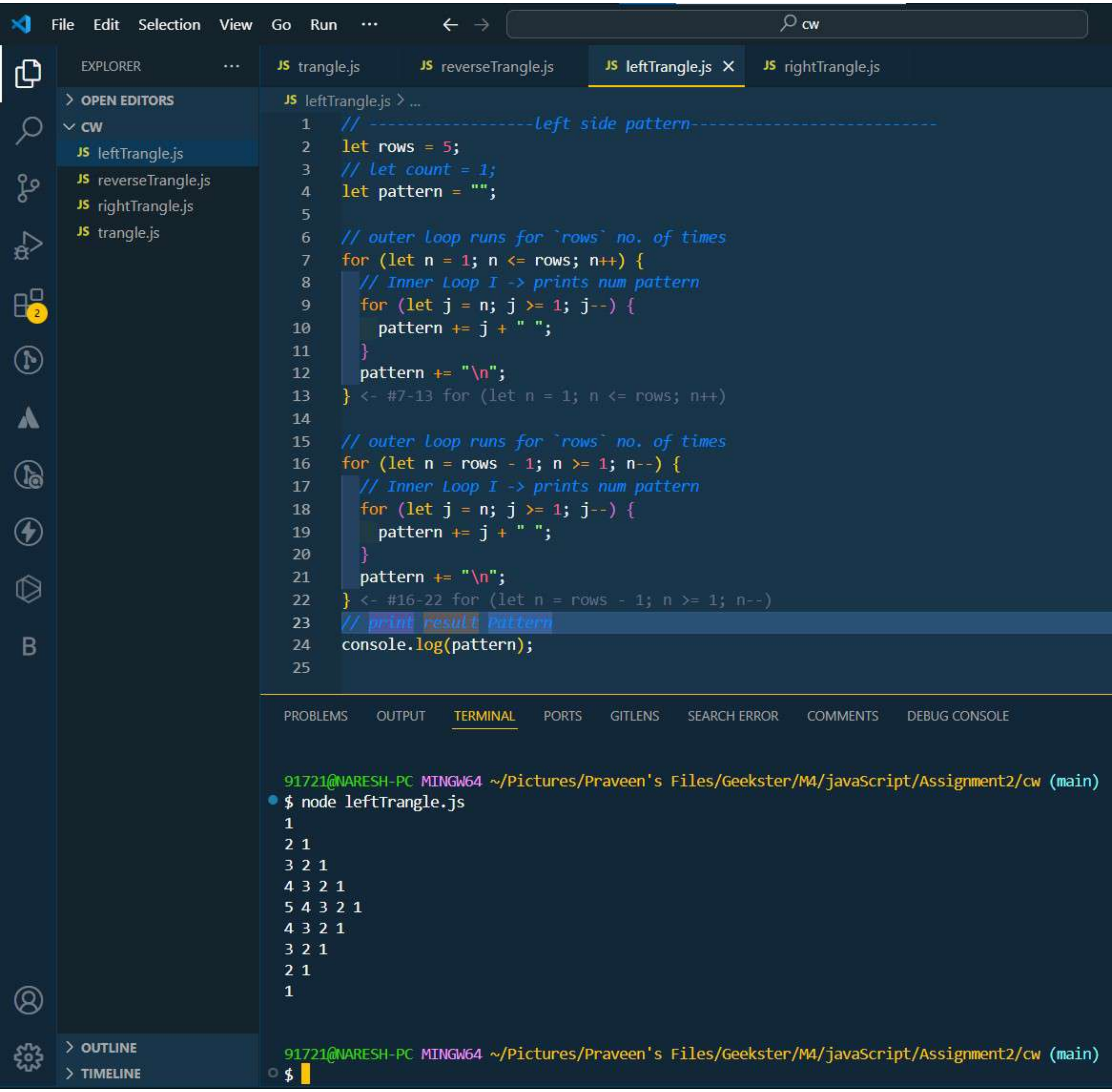
```javascript
// ------------------------Right side pattern --------------------
let rows = 5;
let pattern = "";

// outer loop runs for `rows` no. of times
for (let n = 1; n <= rows; n++) {
  // Inner Loop I -> prints spaces
  for (let space = 1; space <= rows - n; space++) {
    pattern += "  ";
  }

  //  Inner Loop II -> prints num pattern
  for (let j = 1; j <= n; j++) {
    pattern += j + " ";
  }
  pattern += "\n";
} <- #6-17 for (let n = 1; n <= rows; n++)

// outer loop runs for `rows` no. of times
for (let n = rows - 1; n >= 1; n--) {
  // Inner Loop I -> prints spaces
  for (let space = 1; space <= rows - n; space++) {
    pattern += "  ";
  }

  //  Inner Loop II -> prints num pattern
  for (let j = 1; j <= n; j++) {
    pattern += j + " ";
  }
  pattern += "\n";
} <- #20-31 for (let n = rows - 1; n >= 1; n--)
// print result Pattern
console.log(pattern);

```

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ node rightTrangle.js
        1
      1 2
    1 2 3
  1 2 3 4
1 2 3 4 5
  1 2 3 4
    1 2 3
      1 2
        1

91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ 
```

Q3.) Write a program in javascript to create a pattern like this



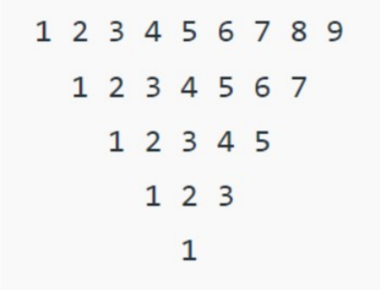
```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```



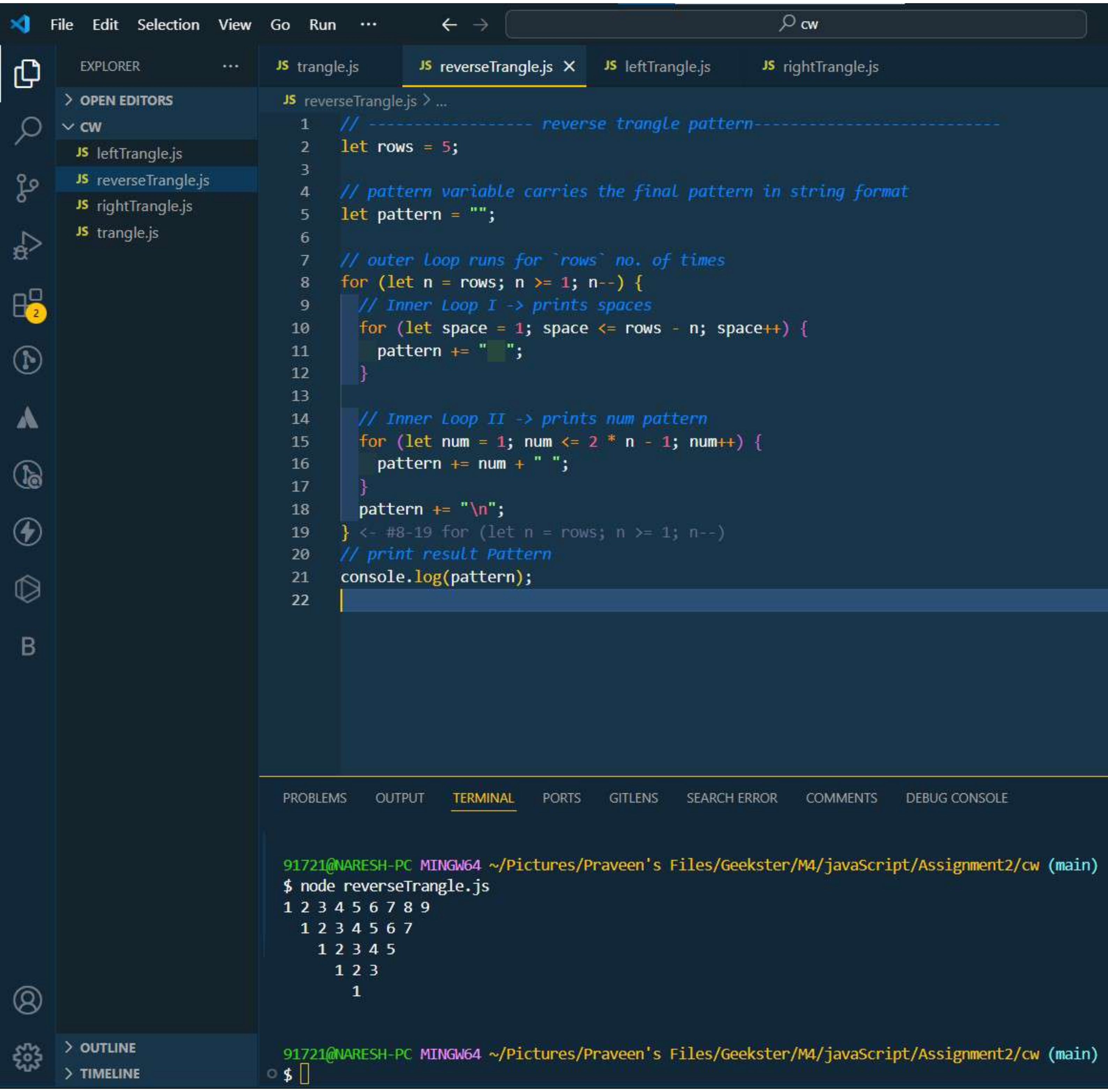
```javascript
// ------------------left side pattern--------------------------
let rows = 5;
// let count = 1;
let pattern = "";

// outer loop runs for `rows` no. of times
for (let n = 1; n <= rows; n++) {
  // Inner Loop I -> prints num pattern
  for (let j = n; j >= 1; j--) {
    pattern += j + " ";
  }
  pattern += "\n";
} <- #7-13 for (let n = 1; n <= rows; n++)

// outer loop runs for `rows` no. of times
for (let n = rows - 1; n >= 1; n--) {
  // Inner Loop I -> prints num pattern
  for (let j = n; j >= 1; j--) {
    pattern += j + " ";
  }
  pattern += "\n";
} <- #16-22 for (let n = rows - 1; n >= 1; n--)
// print result Pattern
console.log(pattern);
```

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ node leftTrangle.js
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1


91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ 
```

Q4.) Write a program in javascript to create a pattern like this



```
1 2 3 4 5 6 7 8 9
  1 2 3 4 5 6 7
    1 2 3 4 5
      1 2 3
        1
```



```javascript
// ----------------- reverse trangle pattern-------------------------
let rows = 5;

// pattern variable carries the final pattern in string format
let pattern = "";

// outer loop runs for `rows` no. of times
for (let n = rows; n >= 1; n--) {
  // Inner Loop I -> prints spaces
  for (let space = 1; space <= rows - n; space++) {
    pattern += "  ";
  }

  // Inner Loop II -> prints num pattern
  for (let num = 1; num <= 2 * n - 1; num++) {
    pattern += num + " ";
  }
  pattern += "\n";
} <- #8-19 for (let n = rows; n >= 1; n--)
// print result Pattern
console.log(pattern);
```

```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ node reverseTrangle.js
1 2 3 4 5 6 7 8 9
  1 2 3 4 5 6 7
    1 2 3 4 5
      1 2 3
        1


91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javaScript/Assignment2/cw (main)
$ 
```