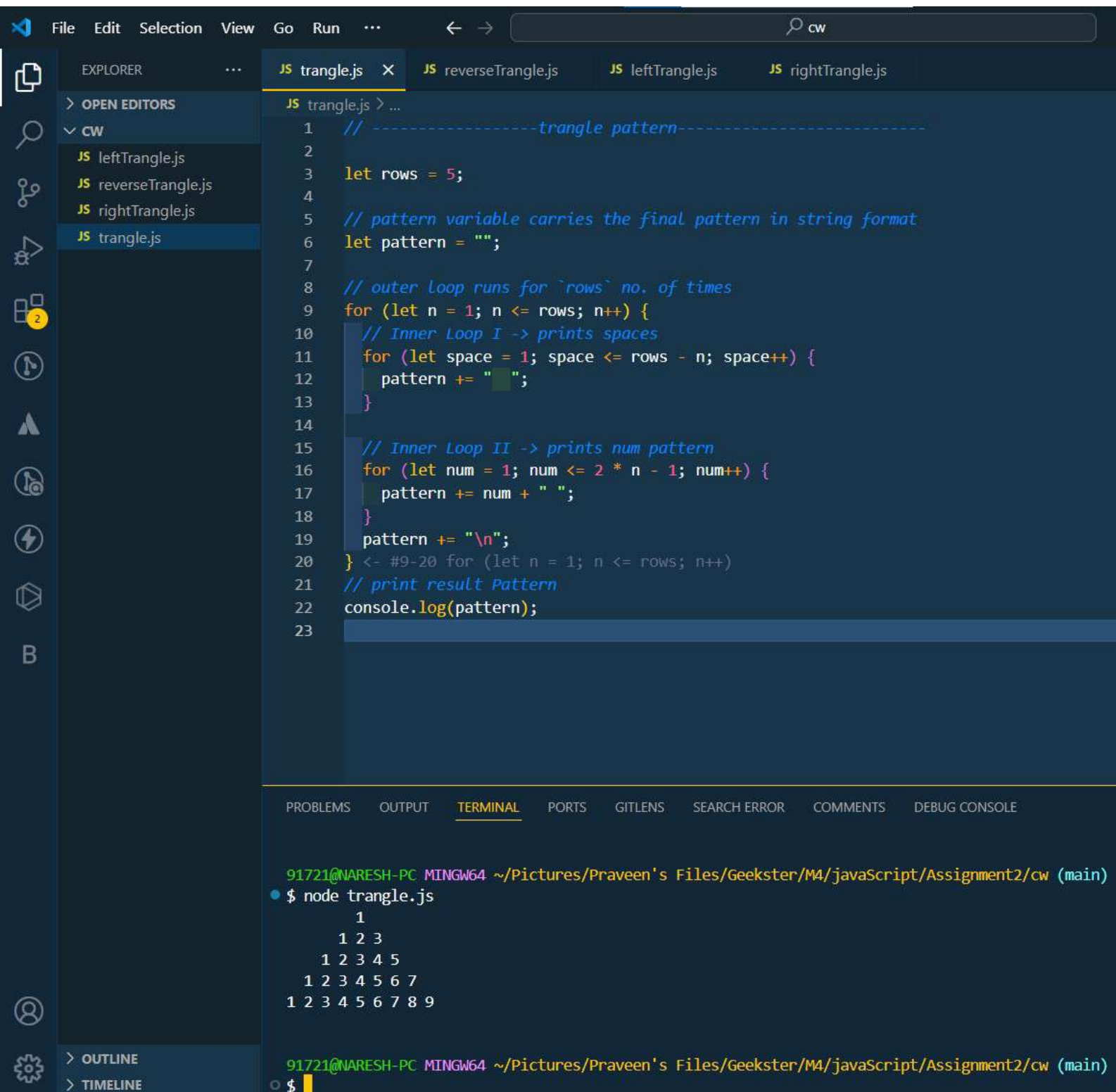


Q1.) Write a program in javascript to create a pattern like this

```
      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
```



```
File Edit Selection View Go Run ... < -> cw
```

```
EXPLORER ... JS trangle.js X JS reverseTrangle.js JS leftTrangle.js JS rightTrangle.js
```

```
> OPEN EDITORS
v CW
JS leftTrangle.js
JS reverseTrangle.js
JS rightTrangle.js
JS trangle.js
```

```
JS trangle.js > ...
1 // -----trangle pattern-----
2
3 let rows = 5;
4
5 // pattern variable carries the final pattern in string format
6 let pattern = "";
7
8 // outer loop runs for `rows` no. of times
9 for (let n = 1; n <= rows; n++) {
10 // Inner Loop I -> prints spaces
11   for (let space = 1; space <= rows - n; space++) {
12     pattern += " ";
13   }
14
15   // Inner Loop II -> prints num pattern
16   for (let num = 1; num <= 2 * n - 1; num++) {
17     pattern += num + " ";
18   }
19   pattern += "\n";
20 } <= #9-20 for (let n = 1; n <= rows; n++)
21 // print result Pattern
22 console.log(pattern);
23
```

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

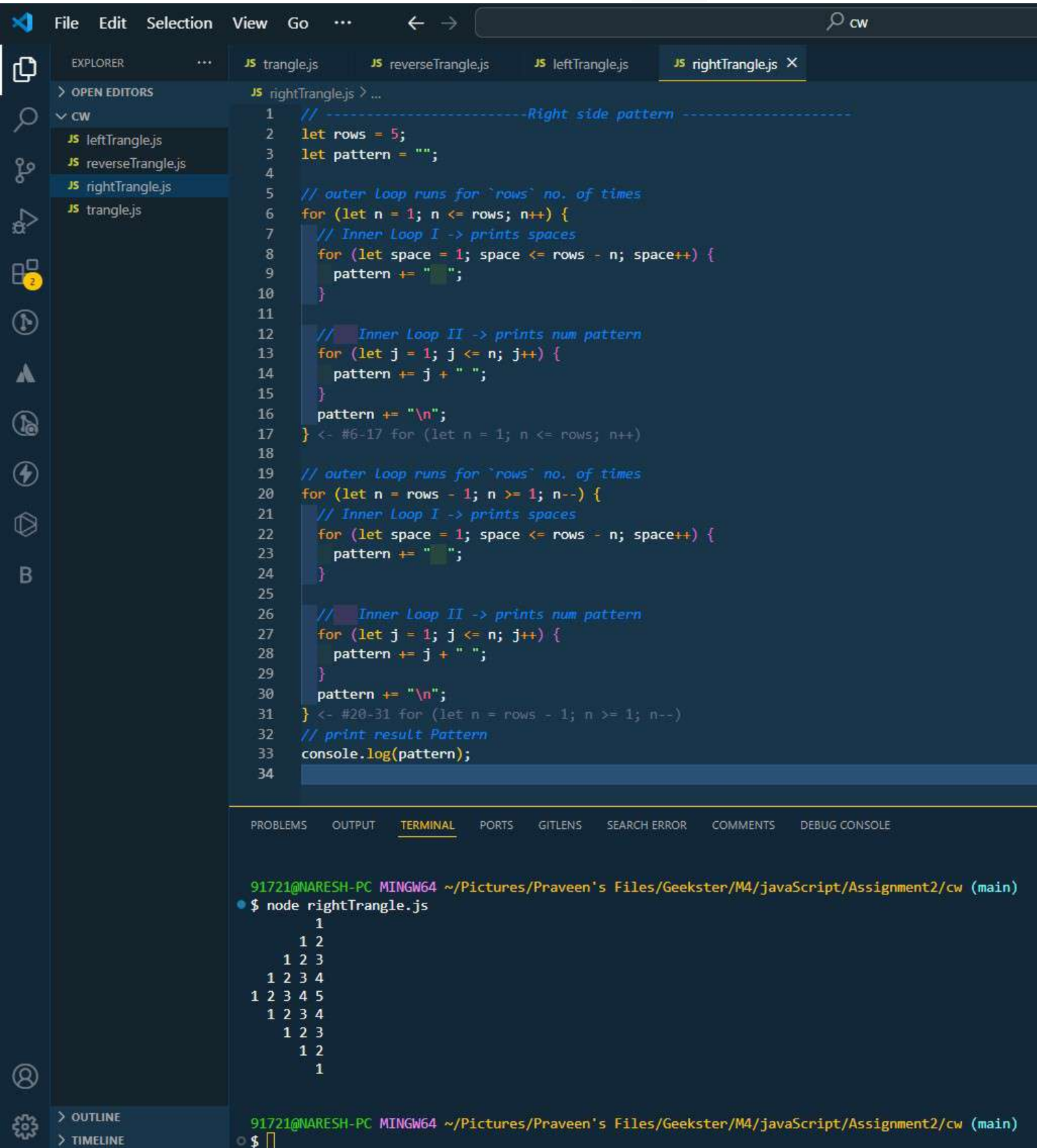
```
91721@NARESH-PC MINGW64 ~/Pictures/Praveen's Files/Geekster/M4/javascript/Assignment2/cw (main)
• $ node trangle.js
      1
    1 2 3
  1 2 3 4 5
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9

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

```
> OUTLINE
> TIMELINE
```

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
```



```
File Edit Selection View Go ... < -> cw
```

```
EXPLORER
```

```
> OPEN EDITORS
```

```
✓ CW
```

```
JS leftTrangle.js
```

```
JS reverseTrangle.js
```

```
JS rightTrangle.js
```

```
JS trangle.js
```

```
JS rightTrangle.js > ...
```

```
1 // -----Right side pattern -----
```

```
2 let rows = 5;
```

```
3 let pattern = "";
```

```
4
```

```
5 // outer loop runs for `rows` no. of times
```

```
6 for (let n = 1; n <= rows; n++) {
```

```
7   // Inner Loop I -> prints spaces
```

```
8   for (let space = 1; space <= rows - n; space++) {
```

```
9     pattern += " ";
```

```
10  }
```

```
11
```

```
12 // Inner loop II -> prints num pattern
```

```
13 for (let j = 1; j <= n; j++) {
```

```
14   pattern += j + " ";
```

```
15 }
```

```
16 pattern += "\n";
```

```
17 } <- #6-17 for (let n = 1; n <= rows; n++)
```

```
18
```

```
19 // outer loop runs for `rows` no. of times
```

```
20 for (let n = rows - 1; n >= 1; n--) {
```

```
21   // Inner Loop I -> prints spaces
```

```
22   for (let space = 1; space <= rows - n; space++) {
```

```
23     pattern += " ";
```

```
24   }
```

```
25
```

```
26 // Inner loop II -> prints num pattern
```

```
27 for (let j = 1; j <= n; j++) {
```

```
28   pattern += j + " ";
```

```
29 }
```

```
30 pattern += "\n";
```

```
31 } <- #20-31 for (let n = rows - 1; n >= 1; n--)
```

```
32 // print result Pattern
```

```
33 console.log(pattern);
```

```
34
```

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

```
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
```

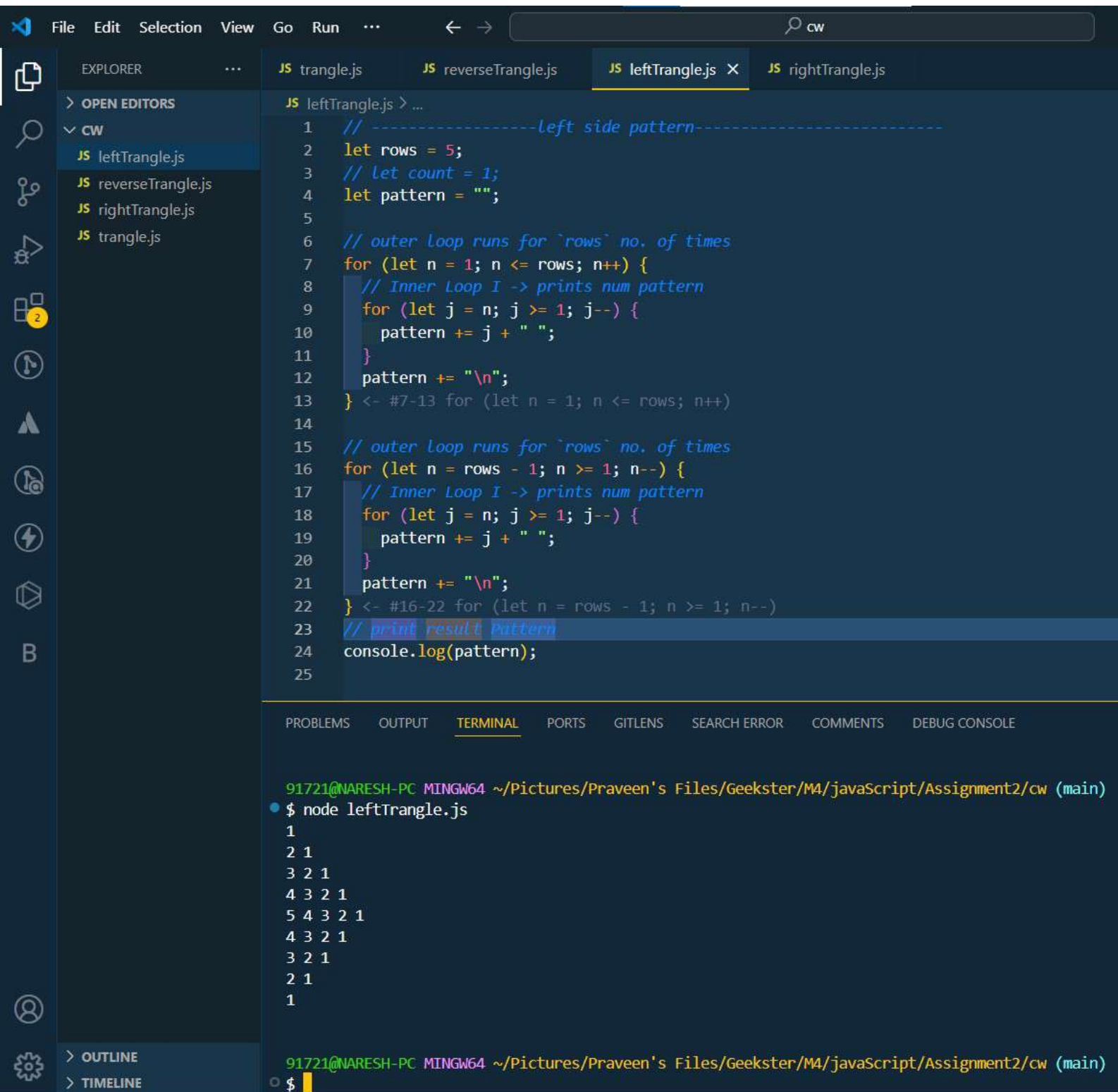
```
1
```

```
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
```



```
File Edit Selection View Go Run ... < -> cw
```

EXPLORER

OPEN EDITORS

CW

- JS leftTriangle.js
- JS reverseTriangle.js
- JS rightTriangle.js
- JS trangle.js

JS leftTriangle.js > ...

```
1 // -----left side pattern-----
2 let rows = 5;
3 // let count = 1;
4 let pattern = "";
5
6 // outer loop runs for `rows` no. of times
7 for (let n = 1; n <= rows; n++) {
8   // Inner Loop I -> prints num pattern
9   for (let j = n; j >= 1; j--) {
10     pattern += j + " ";
11   }
12   pattern += "\n";
13 } <- #7-13 for (let n = 1; n <= rows; n++)
14
15 // outer loop runs for `rows` no. of times
16 for (let n = rows - 1; n >= 1; n--) {
17   // Inner Loop I -> prints num pattern
18   for (let j = n; j >= 1; j--) {
19     pattern += j + " ";
20   }
21   pattern += "\n";
22 } <- #16-22 for (let n = rows - 1; n >= 1; n--)
23 // print result pattern
24 console.log(pattern);
25
```

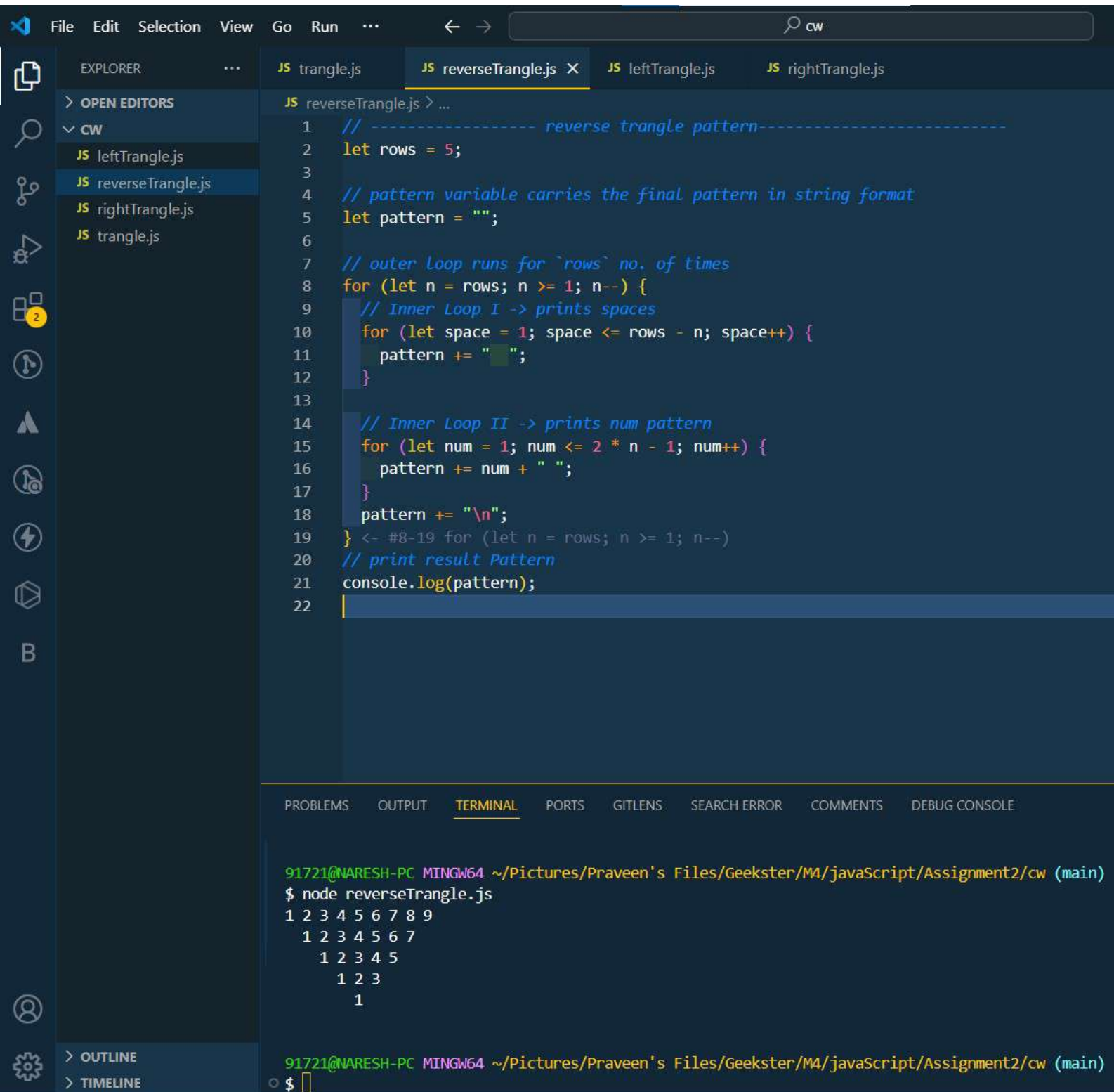
PROBLEMS OUTPUT TERMINAL PORTS GITLENS SEARCH ERROR COMMENTS DEBUG CONSOLE

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



The screenshot shows the Visual Studio Code editor with the file `reverseTriangle.js` open. The code defines a function to generate a reverse triangle pattern. The pattern consists of 5 rows, with the first row having 9 spaces and 1 number, and the last row having 0 spaces and 5 numbers. The code uses nested loops to calculate the number of spaces and numbers for each row.

```
1 // ----- reverse triangle pattern-----
2 let rows = 5;
3
4 // pattern variable carries the final pattern in string format
5 let pattern = "";
6
7 // outer loop runs for `rows` no. of times
8 for (let n = rows; n >= 1; n--) {
9   // Inner Loop I -> prints spaces
10  for (let space = 1; space <= rows - n; space++) {
11    pattern += " ";
12  }
13
14  // Inner Loop II -> prints num pattern
15  for (let num = 1; num <= 2 * n - 1; num++) {
16    pattern += num + " ";
17  }
18  pattern += "\n";
19 } <- #8-19 for (let n = rows; n >= 1; n--)
20 // print result Pattern
21 console.log(pattern);
22
```

The terminal output shows the pattern generated by the program:

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