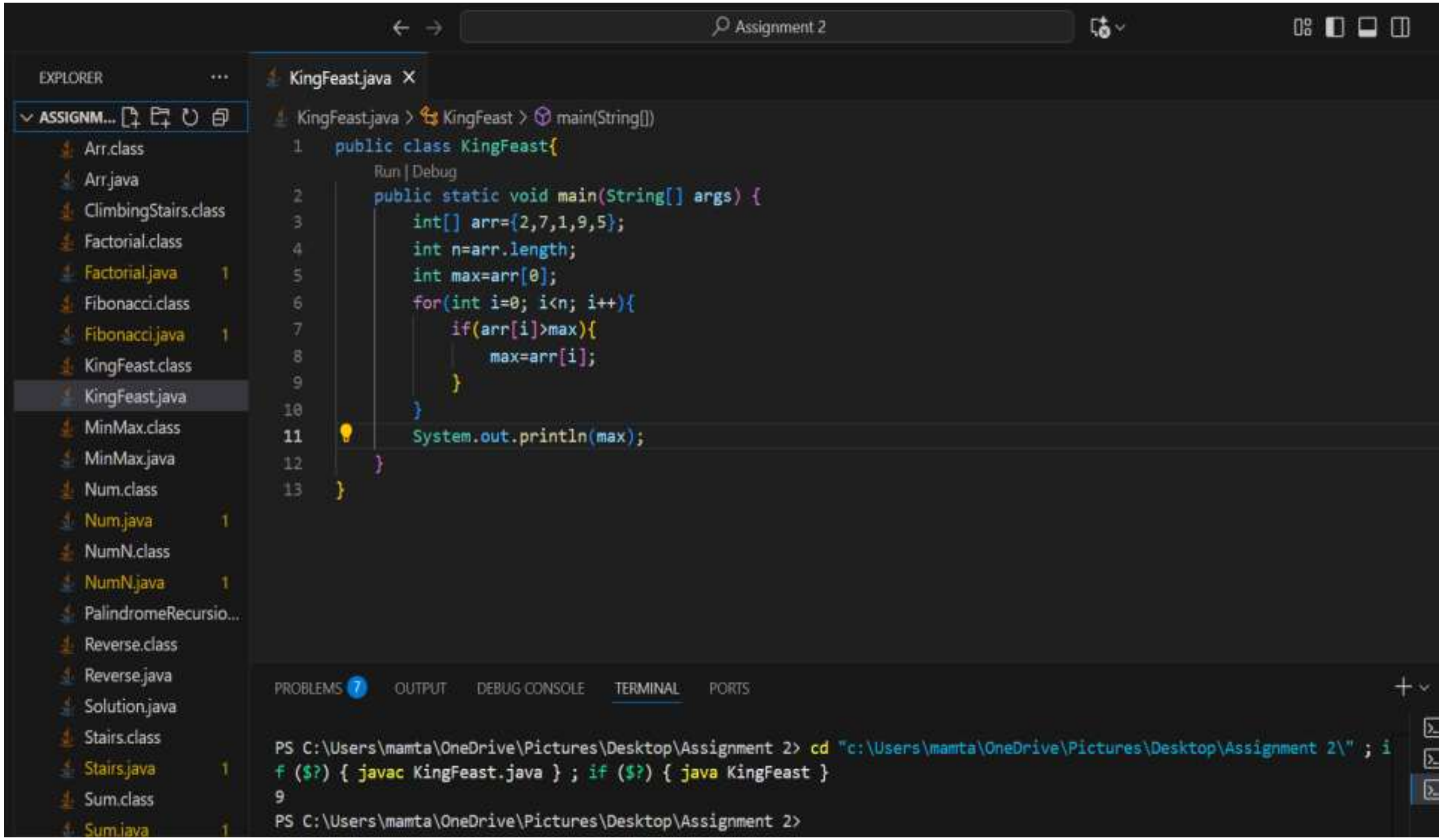
Assignment-3

DSA-Problems

(Date: 06 October, 2025)

- Name:- Abhay Fulara
- Admission no:- 24SCSE1180211
- Section:- 34

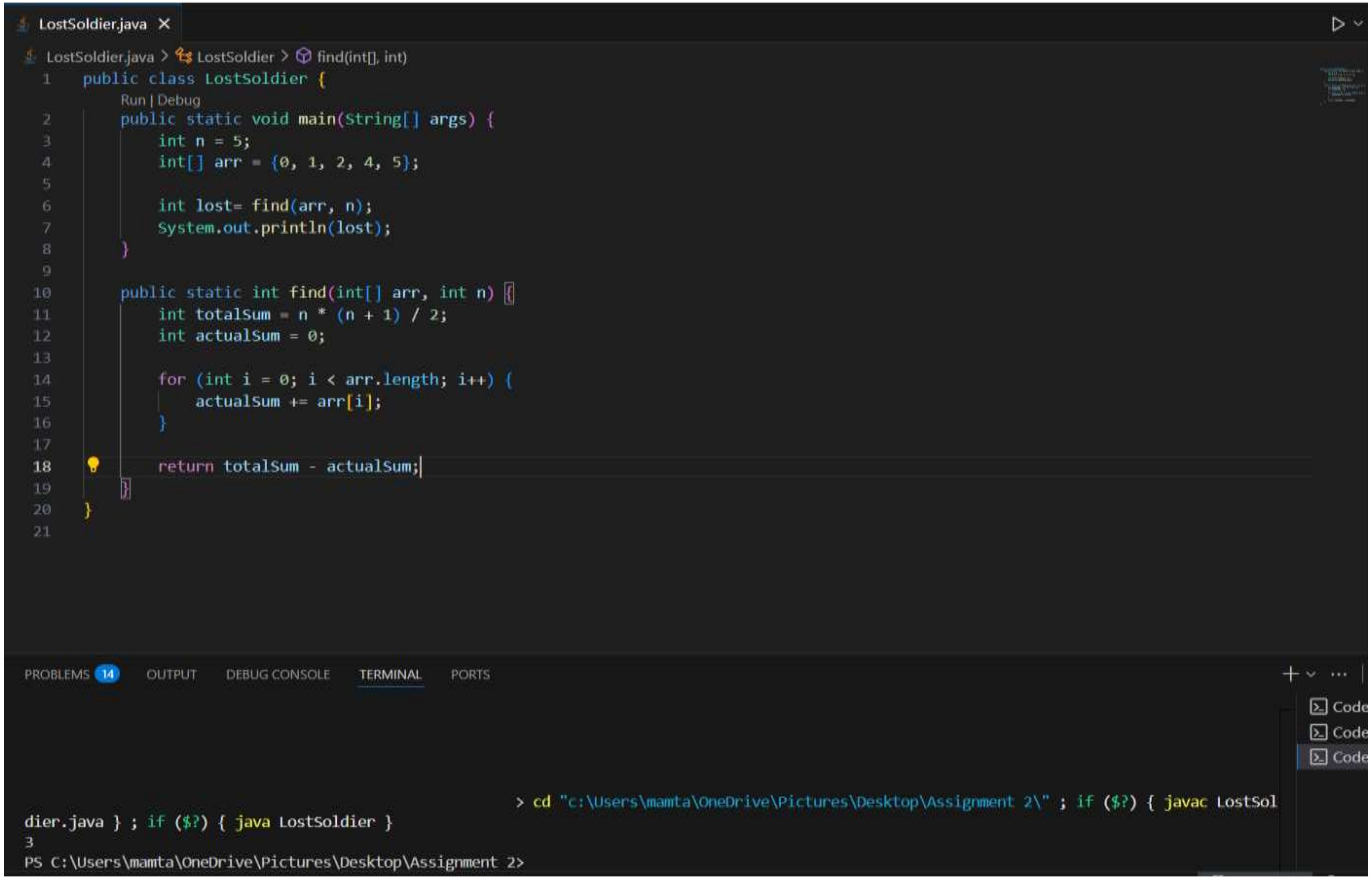
Q1. The King's Feast



```
public class KingFeast{
    public static void main(String[] args) {
        int[] arr={2,7,1,9,5};
        int n=arr.length;
        int max=arr[0];
        for(int i=0; i<n; i++){
            if(arr[i]>max){
                max=arr[i];
            }
        }
        System.out.println(max);
    }
}
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; i
f ($?) { javac KingFeast.java } ; if ($?) { java KingFeast }
9
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

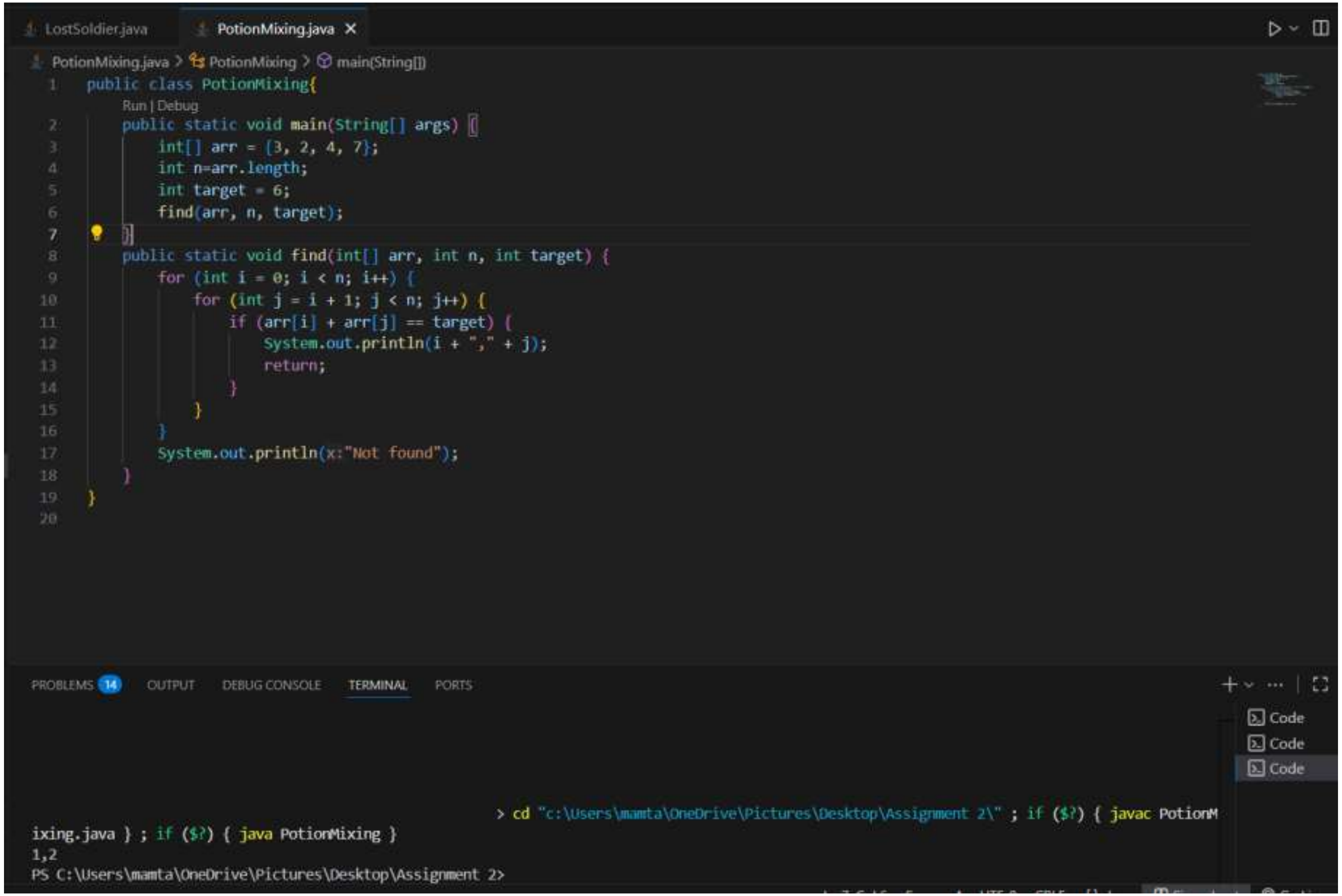
Q2. The Lost Soldier



```java
public class LostSoldier {
    public static void main(String[] args) {
        int n = 5;
        int[] arr = {0, 1, 2, 4, 5};

        int lost= find(arr, n);
        System.out.println(lost);
    }

    public static int find(int[] arr, int n) {
        int totalSum = n * (n + 1) / 2;
        int actualSum = 0;

        for (int i = 0; i < arr.length; i++) {
            actualSum += arr[i];
        }

        return totalSum - actualSum;
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac LostSol
dier.java } ; if ($?) { java LostSoldier }
3
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

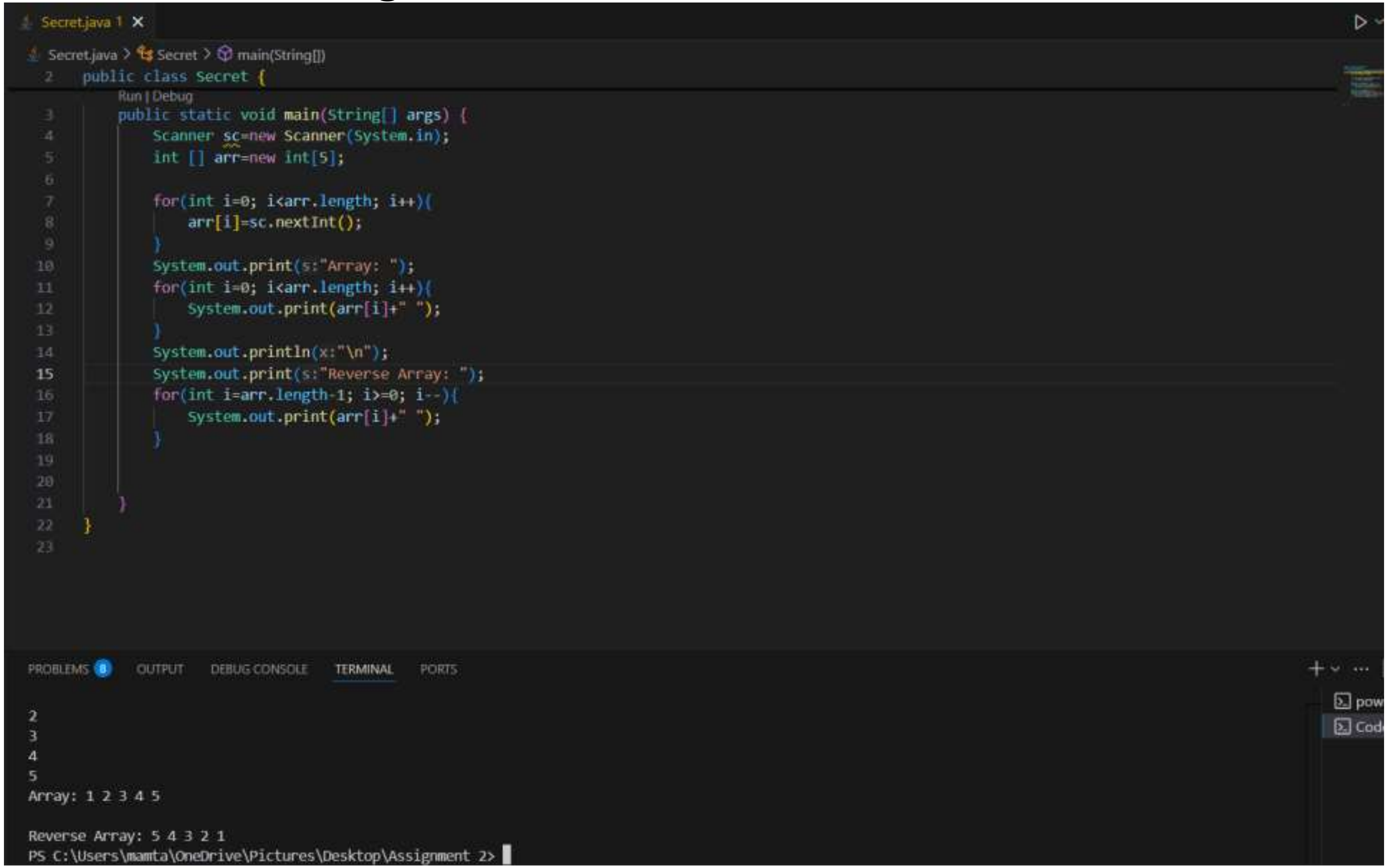
Q3. Potion Mixing (Two Sum)



```java
public class PotionMixing{
    public static void main(String[] args) {
        int[] arr = {3, 2, 4, 7};
        int n=arr.length;
        int target = 6;
        find(arr, n, target);
    }
    public static void find(int[] arr, int n, int target) {
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) {
                if (arr[i] + arr[j] == target) {
                    System.out.println(i + "," + j);
                    return;
                }
            }
        }
        System.out.println(x:"Not found");
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac PotionMixing.java } ; if ($?) { java PotionMixing }
1,2
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q4. The Secret Message



Secret.java 1 ×

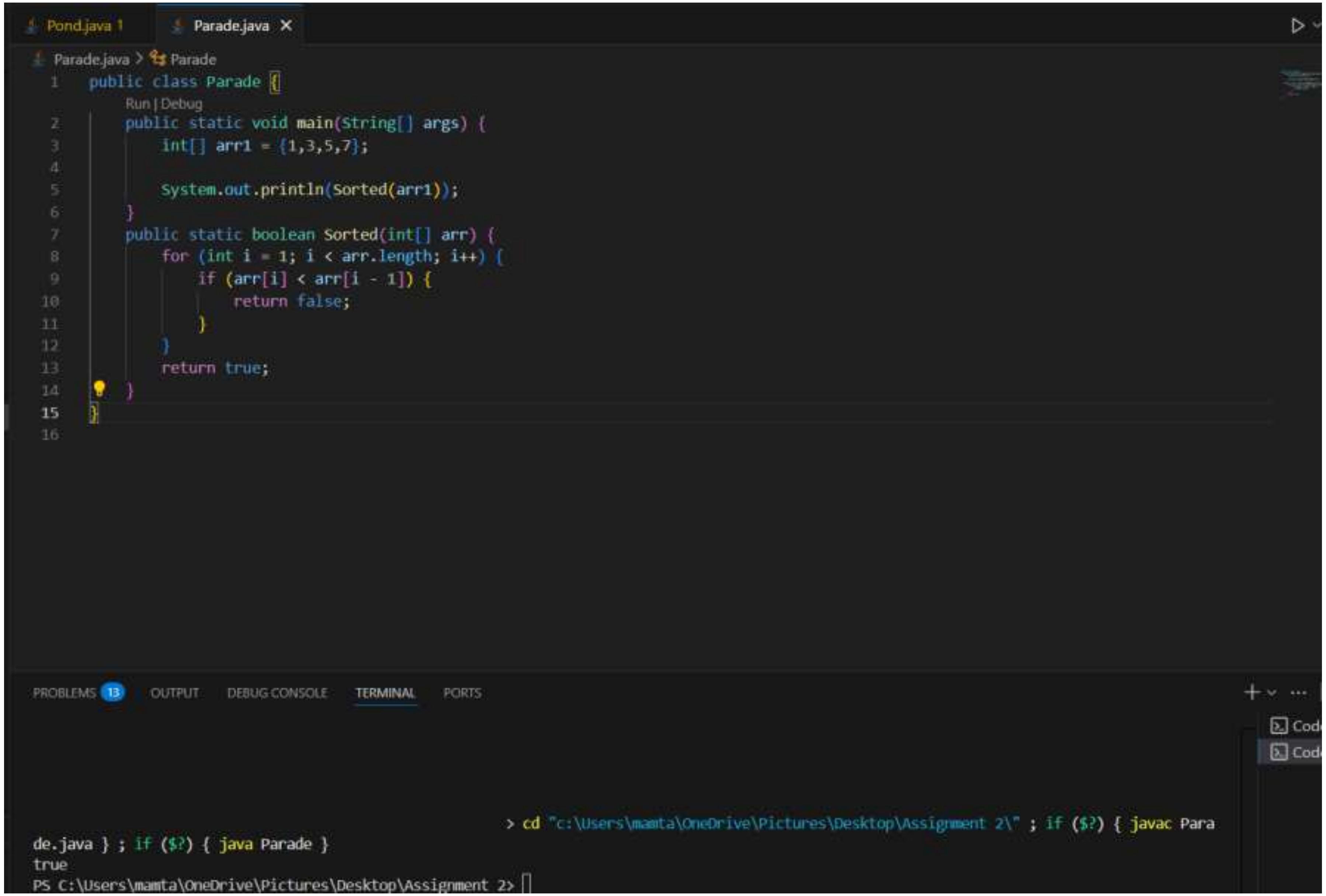
Secret.java > Secret > main(String[])

```java
public class Secret {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int [] arr=new int[5];

        for(int i=0; i<arr.length; i++){
            arr[i]=sc.nextInt();
        }
        System.out.print(s:"Array: ");
        for(int i=0; i<arr.length; i++){
            System.out.print(arr[i]+" ");
        }
        System.out.println(x:"\n");
        System.out.print(s:"Reverse Array: ");
        for(int i=arr.length-1; i>=0; i--){
            System.out.print(arr[i]+" ");
        }


    }
}
```

PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
2
3
4
5
Array: 1 2 3 4 5

Reverse Array: 5 4 3 2 1
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

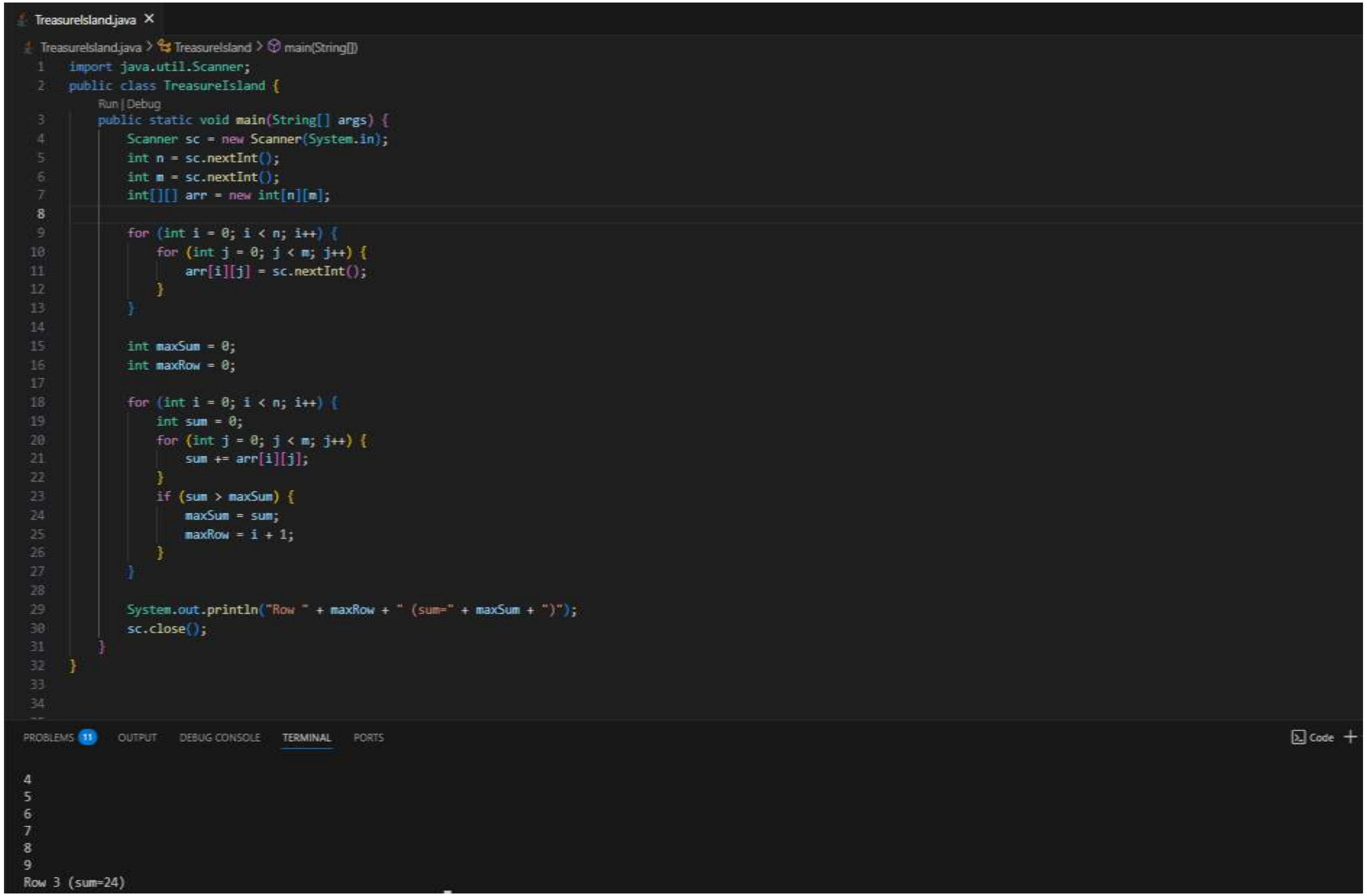
Q5. The King’s Parade



```java
public class Parade {
    public static void main(String[] args) {
        int[] arr1 = {1,3,5,7};

        System.out.println(Sorted(arr1));
    }
    public static boolean Sorted(int[] arr) {
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] < arr[i - 1]) {
                return false;
            }
        }
        return true;
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Para
de.java } ; if ($?) { java Parade }
true
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q6. The Treasure Island



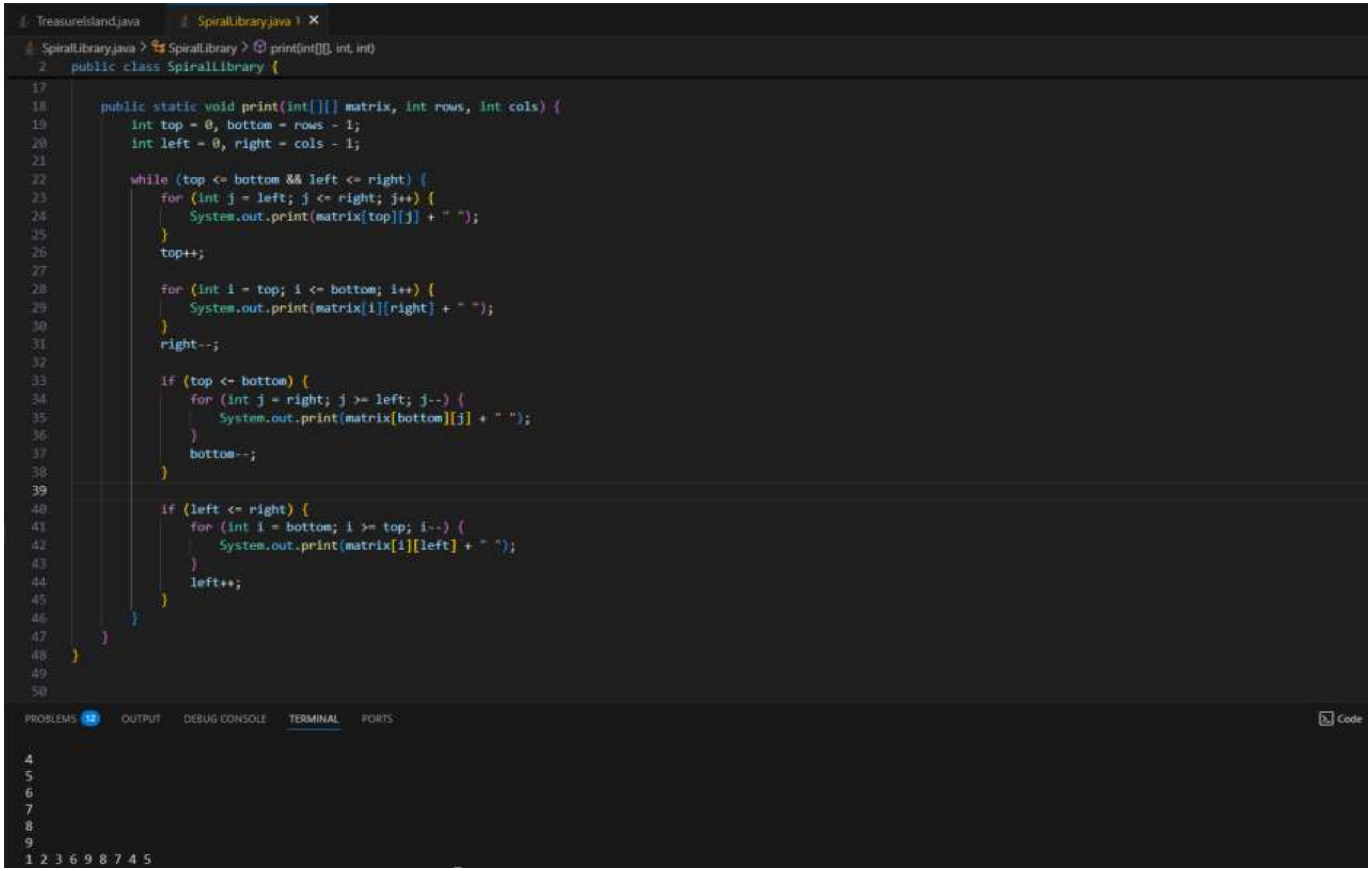
TreasureIsland.java ×

TreasureIsland.java > TreasureIsland > main(String[])

```java
import java.util.Scanner;
public class TreasureIsland {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int m = sc.nextInt();
        int[][] arr = new int[n][m];

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                arr[i][j] = sc.nextInt();
            }
        }

        int maxSum = 0;
        int maxRow = 0;

        for (int i = 0; i < n; i++) {
            int sum = 0;
            for (int j = 0; j < m; j++) {
                sum += arr[i][j];
            }
            if (sum > maxSum) {
                maxSum = sum;
                maxRow = i + 1;
            }
        }

        System.out.println("Row " + maxRow + " (sum=" + maxSum + ")");
        sc.close();
    }
}
```

```
4
5
6
7
8
9
Row 3 (sum=24)
```

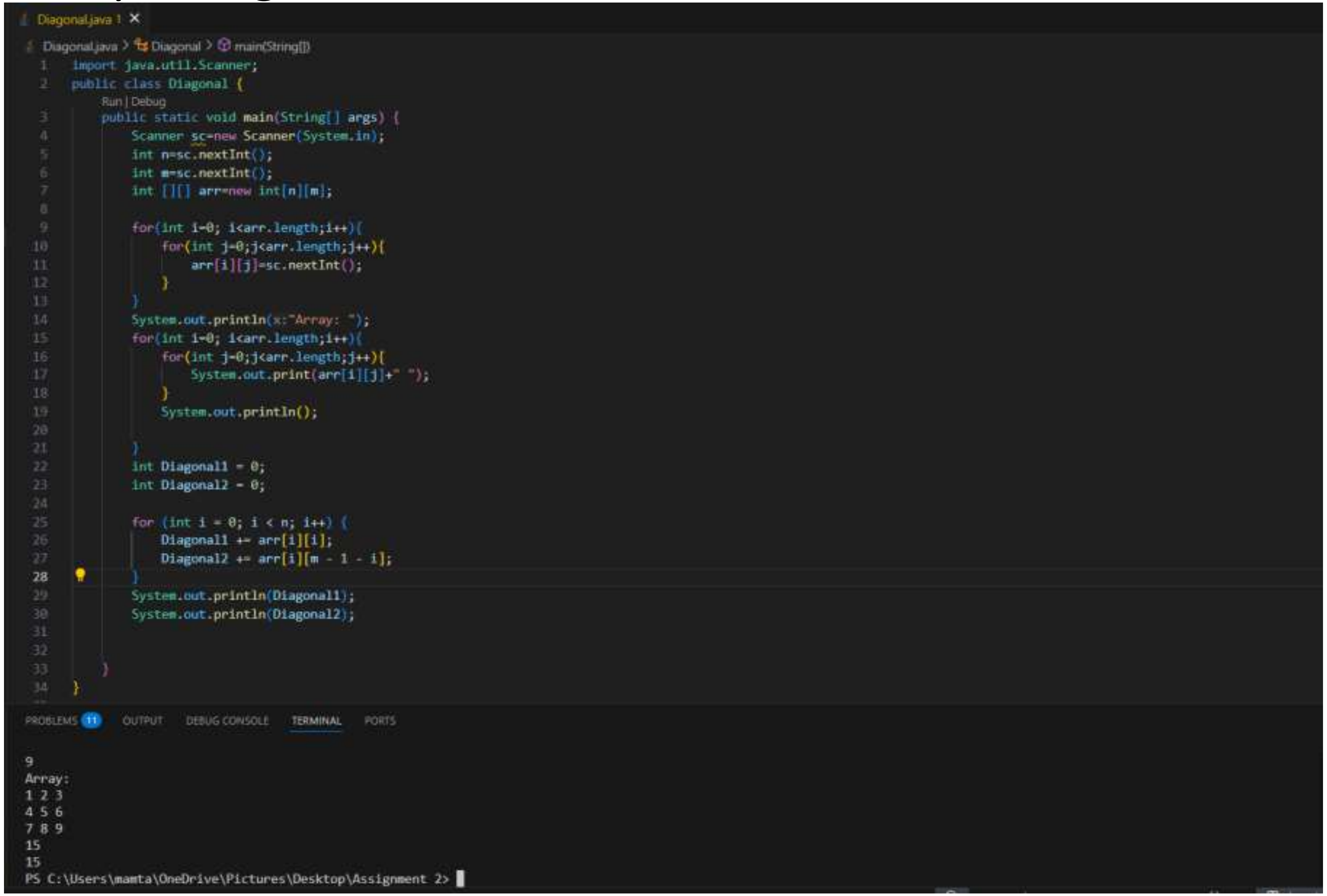
Q7. The Spiral Library



```java
public class SpiralLibrary {

    public static void print(int[][] matrix, int rows, int cols) {
        int top = 0, bottom = rows - 1;
        int left = 0, right = cols - 1;

        while (top <= bottom && left <= right) {
            for (int j = left; j <= right; j++) {
                System.out.print(matrix[top][j] + " ");
            }
            top++;

            for (int i = top; i <= bottom; i++) {
                System.out.print(matrix[i][right] + " ");
            }
            right--;

            if (top <= bottom) {
                for (int j = right; j >= left; j--) {
                    System.out.print(matrix[bottom][j] + " ");
                }
                bottom--;
            }

            if (left <= right) {
                for (int i = bottom; i >= top; i--) {
                    System.out.print(matrix[i][left] + " ");
                }
                left++;
            }
        }
    }
}
```

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

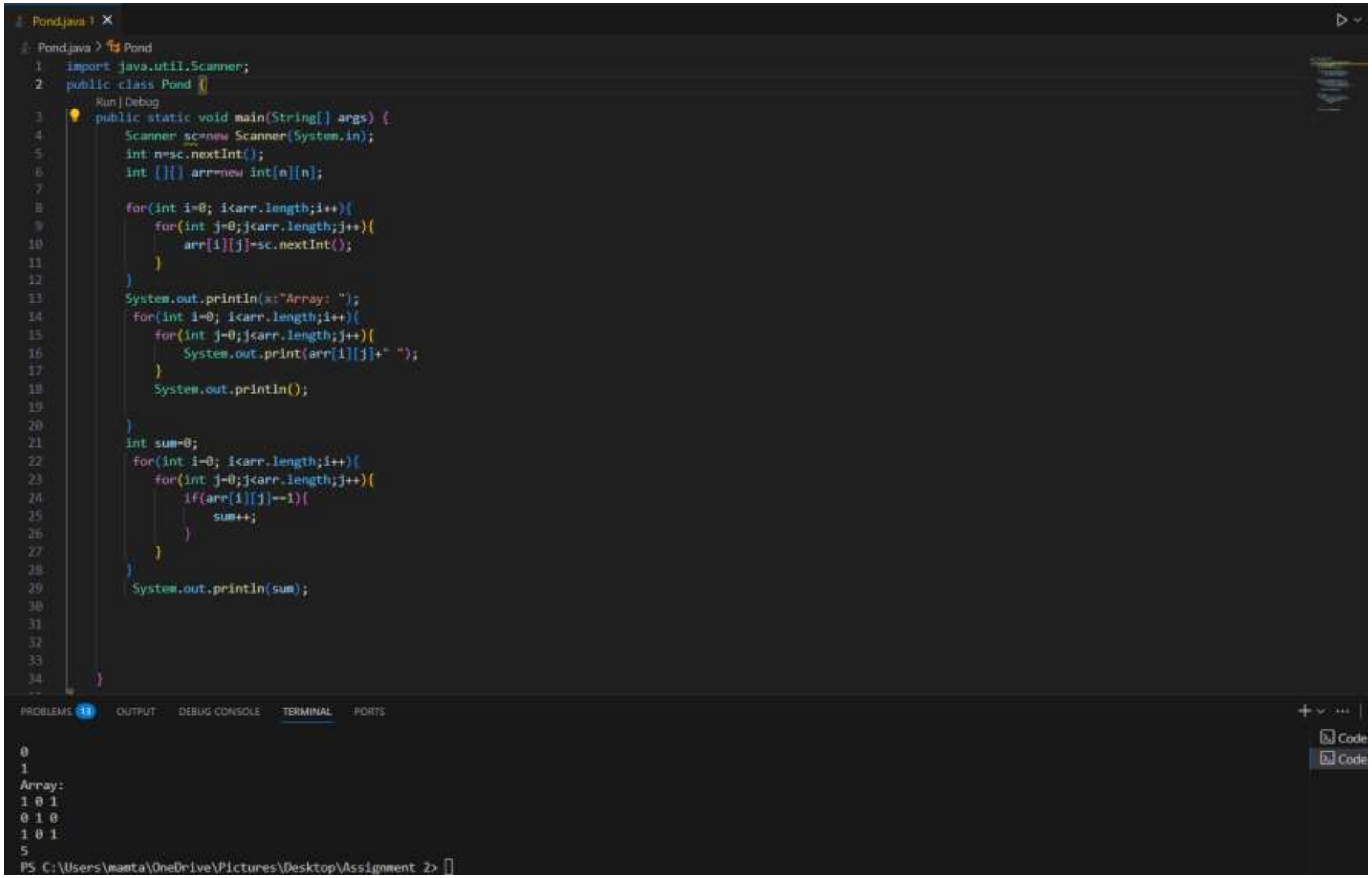
Q8. The Royal Diagonal



```java
import java.util.Scanner;
public class Diagonal {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int m=sc.nextInt();
        int [][] arr=new int[n][m];

        for(int i=0; i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                arr[i][j]=sc.nextInt();
            }
        }
        System.out.println(x:"Array: ");
        for(int i=0; i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();

        }
        int Diagonal1 = 0;
        int Diagonal2 = 0;

        for (int i = 0; i < n; i++) {
            Diagonal1 += arr[i][i];
            Diagonal2 += arr[i][m - 1 - i];
        }
        System.out.println(Diagonal1);
        System.out.println(Diagonal2);


    }
}
```

```
9
Array:
1 2 3
4 5 6
7 8 9
15
15
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> 
```

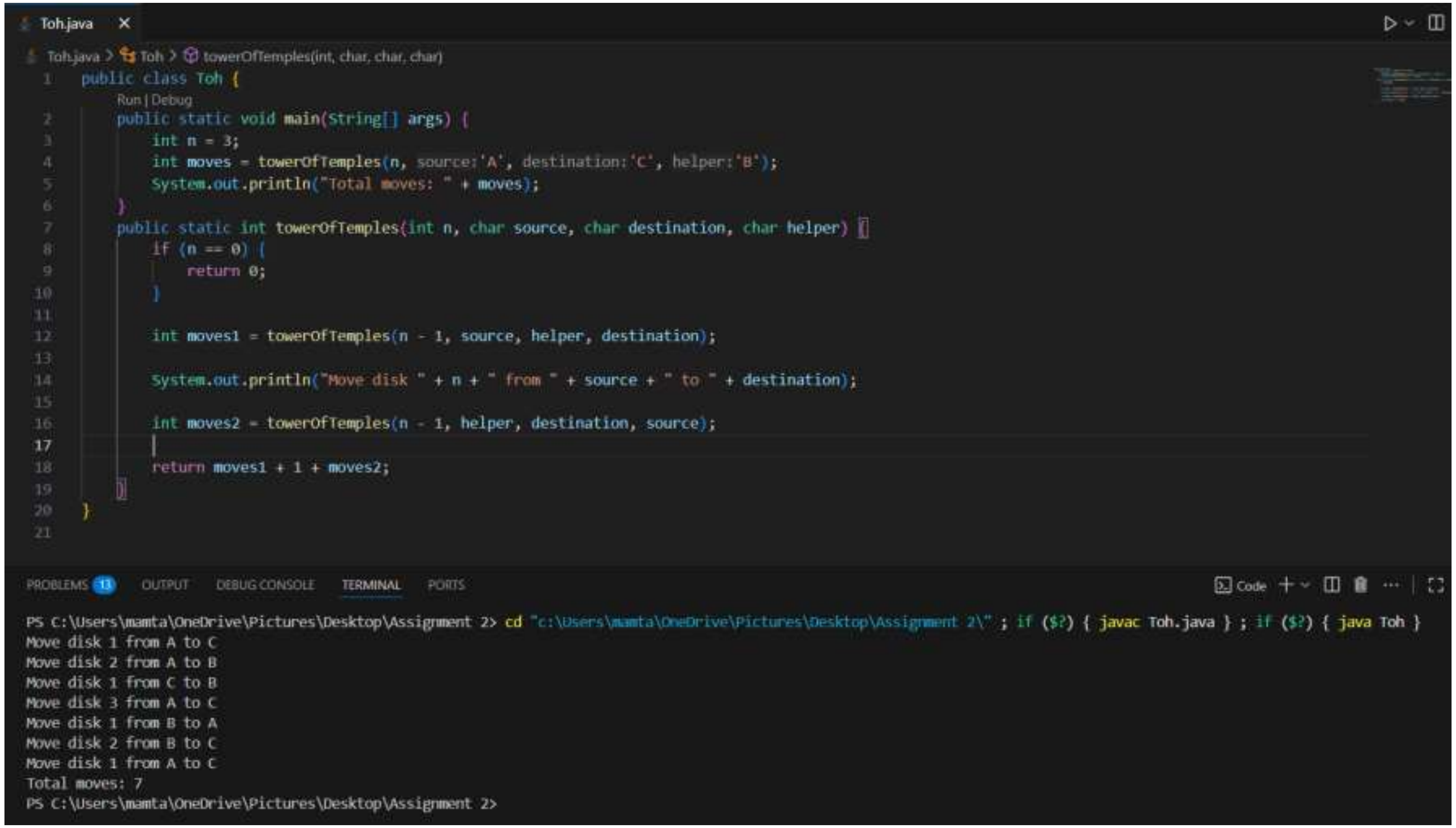
Q10. The Rainwater Pond



```java
import java.util.Scanner;
public class Pond {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int [][] arr=new int[n][n];

        for(int i=0; i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                arr[i][j]=sc.nextInt();
            }
        }
        System.out.println(x:"Array: ");
         for(int i=0; i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                System.out.print(arr[i][j]+" ");
            }
            System.out.println();

        }
        int sum=0;
         for(int i=0; i<arr.length;i++){
            for(int j=0;j<arr.length;j++){
                if(arr[i][j]==1){
                    sum++;
                }
            }
        }
         System.out.println(sum);





    }
```

```
0
1
Array:
1 0 1
0 1 0
1 0 1
5
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> 
```

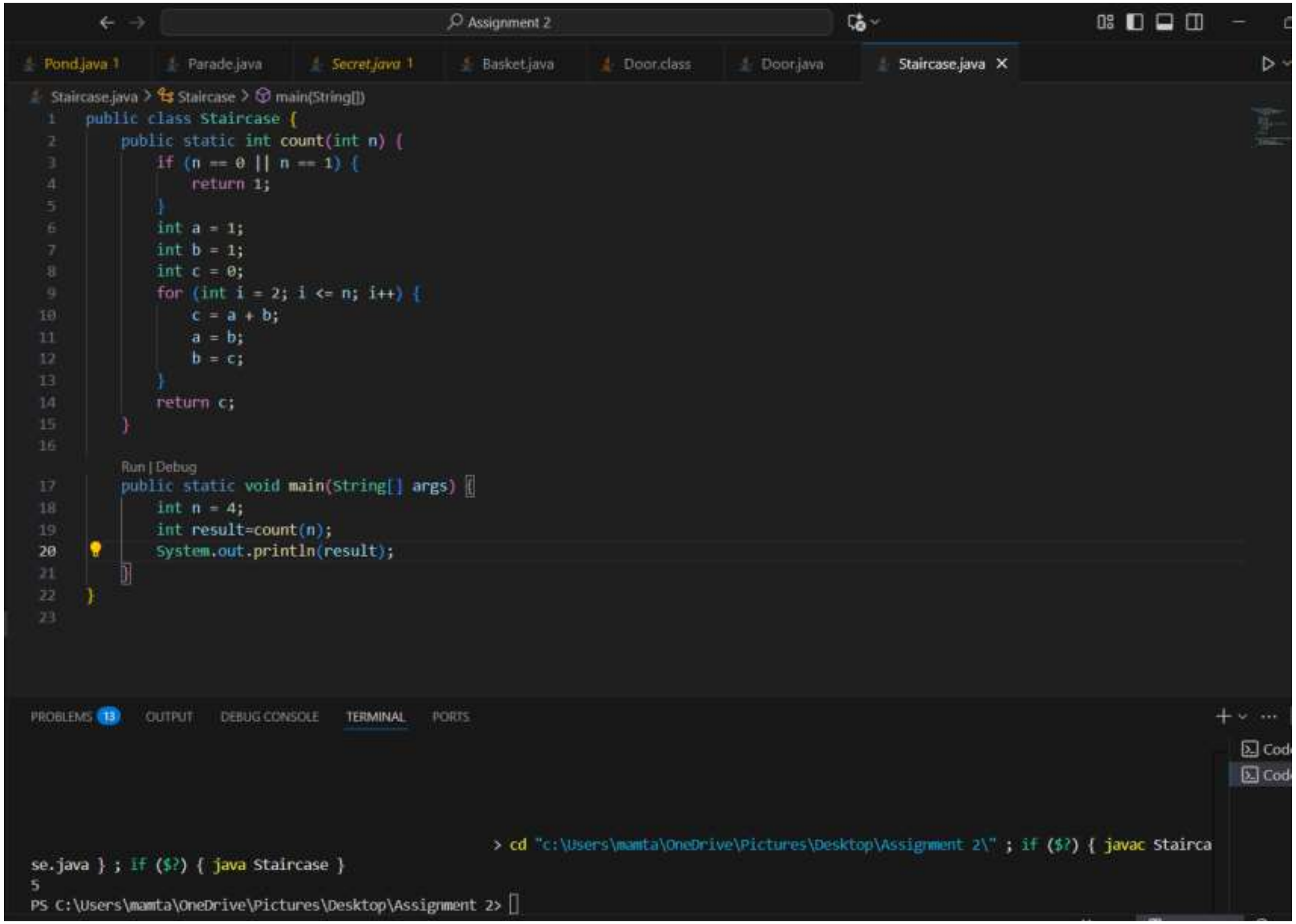
Q11. Tower of Temples (Hanoi)


Toh.java
Toh.java > Toh > towerOfTemples(int, char, char, char)
Run | Debug
PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```java
public class Toh {
    public static void main(String[] args) {
        int n = 3;
        int moves = towerOfTemples(n, source:'A', destination:'C', helper:'B');
        System.out.println("Total moves: " + moves);
    }
    public static int towerOfTemples(int n, char source, char destination, char helper) {
        if (n == 0) {
            return 0;
        }

        int moves1 = towerOfTemples(n - 1, source, helper, destination);

        System.out.println("Move disk " + n + " from " + source + " to " + destination);

        int moves2 = towerOfTemples(n - 1, helper, destination, source);

        return moves1 + 1 + moves2;
    }
}
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Toh.java } ; if ($?) { java Toh }
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total moves: 7
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q12. The Magical Staircase



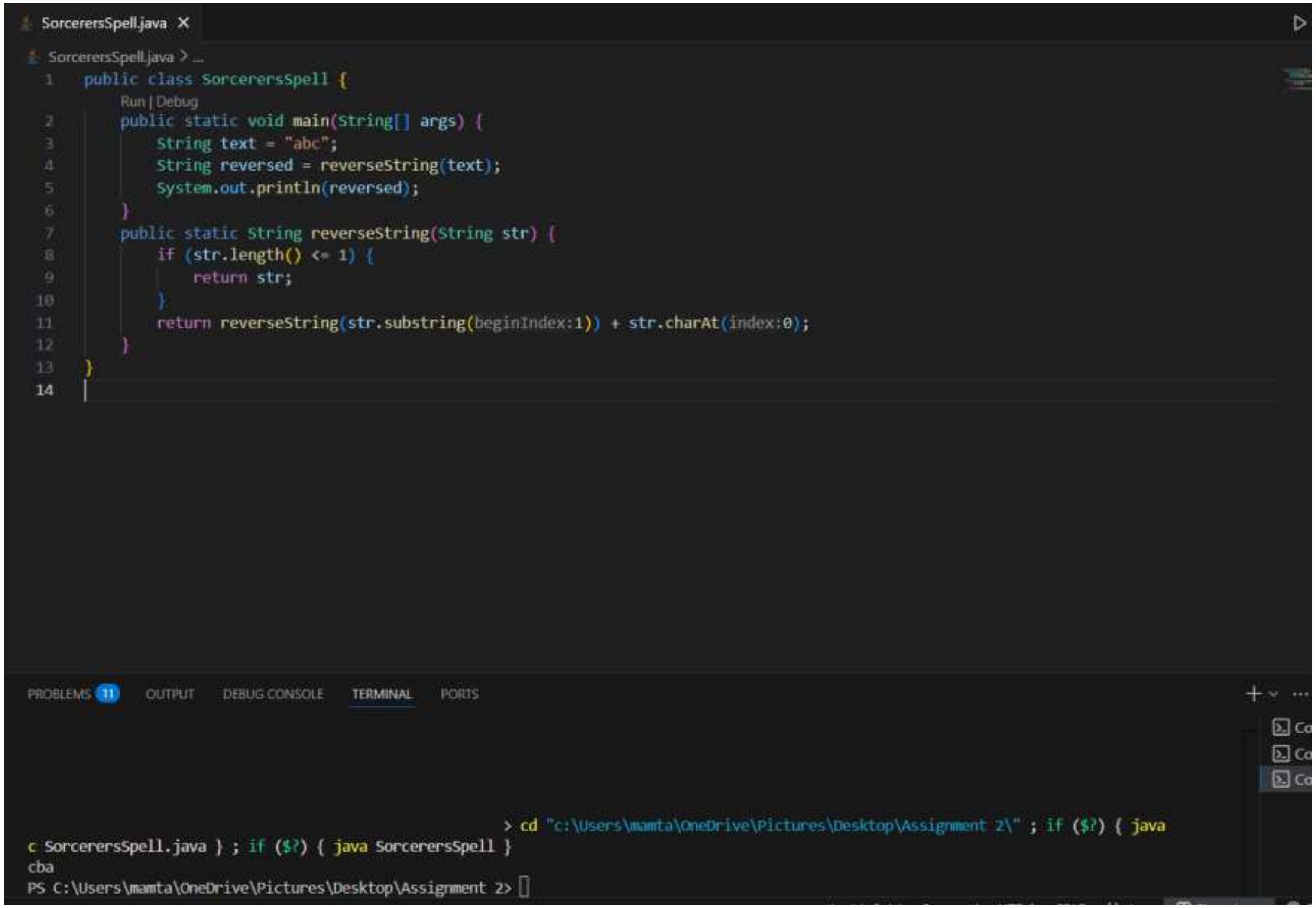
Pond.java 1 | Parade.java | Secret.java 1 | Basket.java | Door.class | Door.java | Staircase.java

Staircase.java > Staircase > main(String[])

```java
public class Staircase {
    public static int count(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        int a = 1;
        int b = 1;
        int c = 0;
        for (int i = 2; i <= n; i++) {
            c = a + b;
            a = b;
            b = c;
        }
        return c;
    }

    public static void main(String[] args) {
        int n = 4;
        int result=count(n);
        System.out.println(result);
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Stairca
se.java } ; if ($?) { java Staircase }
5
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q13. The Sorcerer’s Spell

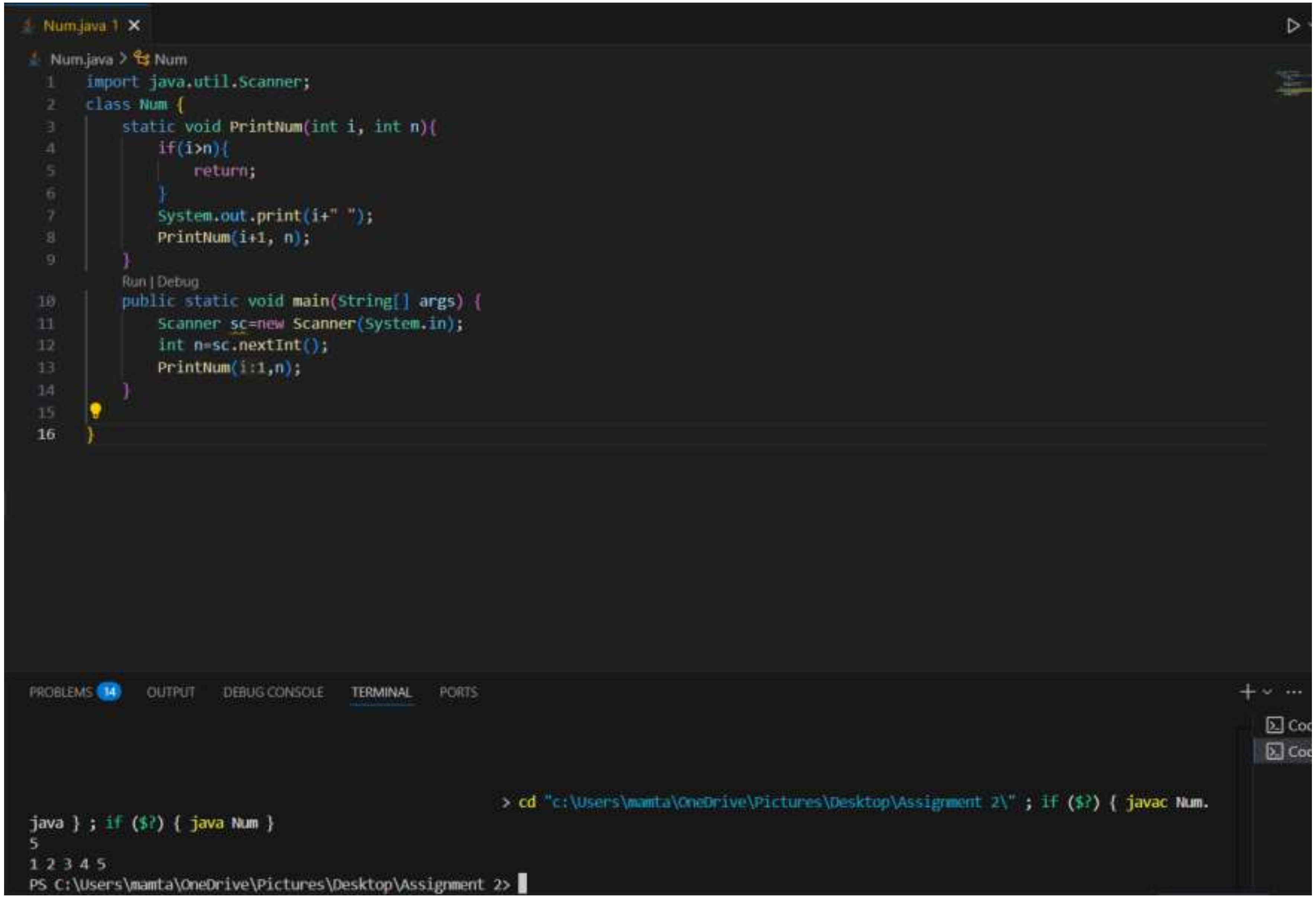


SorcerersSpell.java

```java
public class SorcerersSpell {
    public static void main(String[] args) {
        String text = "abc";
        String reversed = reverseString(text);
        System.out.println(reversed);
    }
    public static String reverseString(String str) {
        if (str.length() <= 1) {
            return str;
        }
        return reverseString(str.substring(beginIndex:1)) + str.charAt(index:0);
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { java
c SorcerersSpell.java } ; if ($?) { java SorcerersSpell }
cba
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

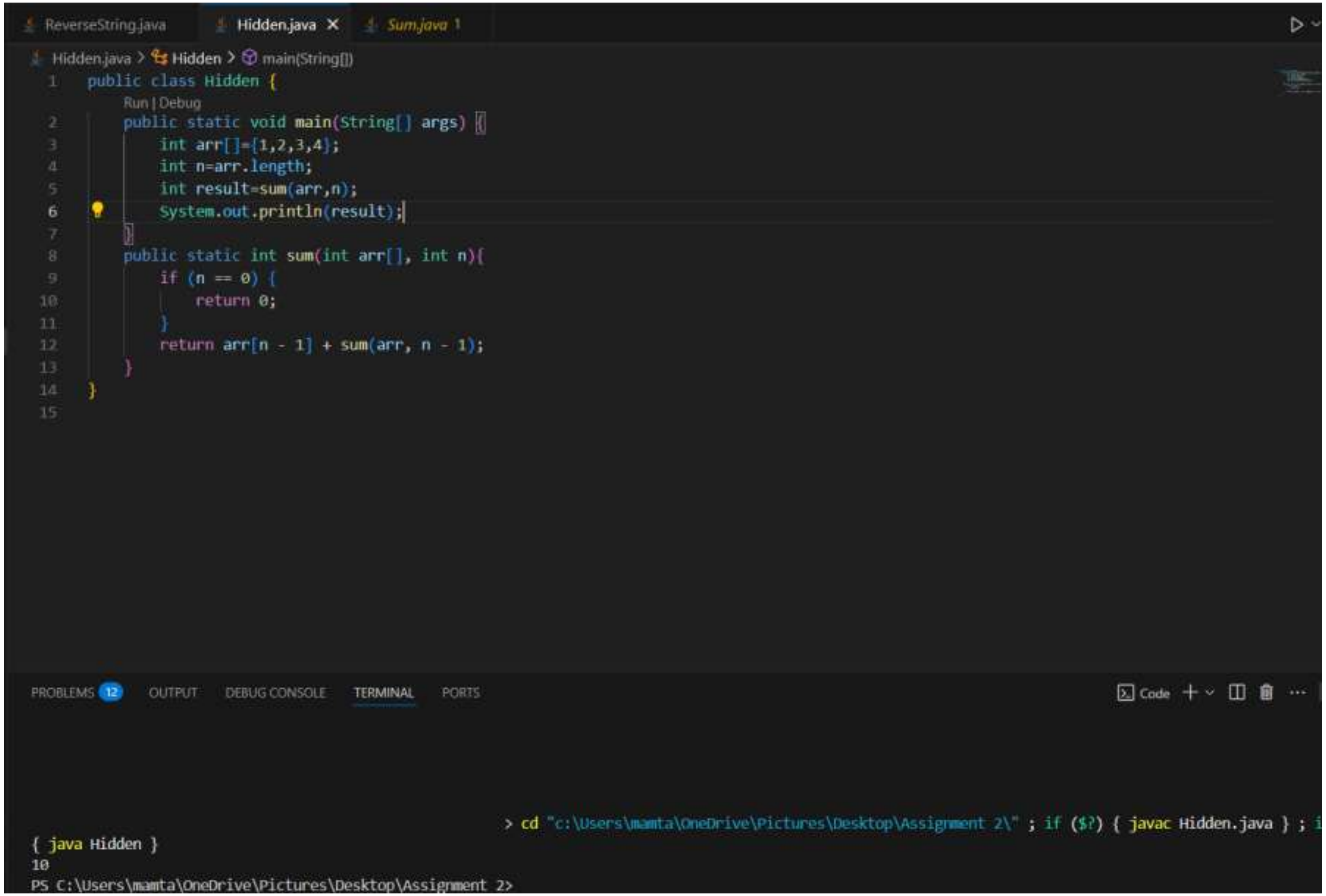
Q14. The Dragon’s Roar



```java
import java.util.Scanner;
class Num {
    static void PrintNum(int i, int n){
        if(i>n){
            return;
        }
        System.out.print(i+" ");
        PrintNum(i+1, n);
    }
    Run | Debug
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        PrintNum(i:1,n);
    }

}
```

```
                                         > cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Num.
java } ; if ($?) { java Num }
5
1 2 3 4 5
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q15. The Hidden Chamber



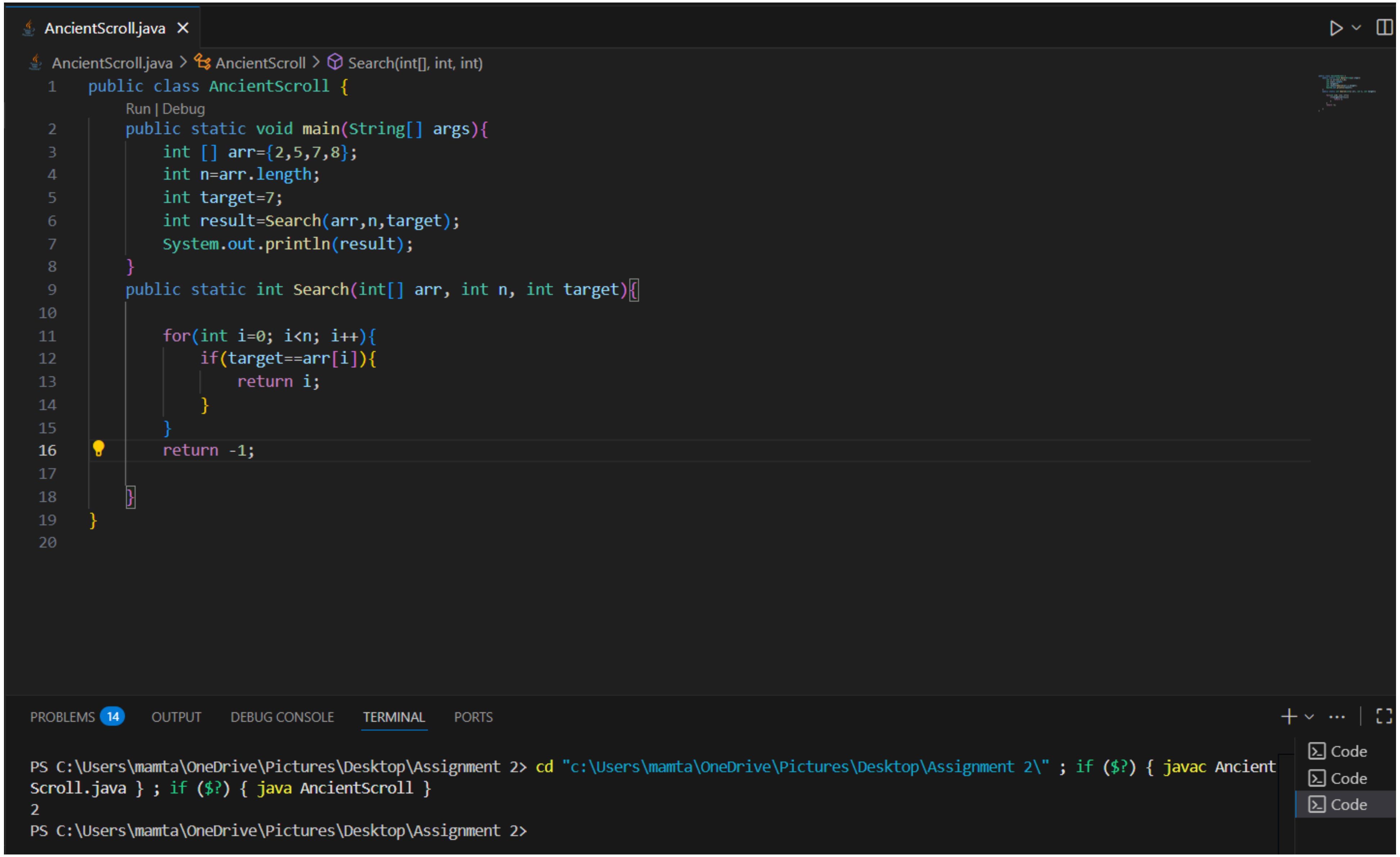
ReverseString.java Hidden.java Sum.java

Hidden.java > Hidden > main(String[])

```java
public class Hidden {
    public static void main(String[] args) {
        int arr[]={1,2,3,4};
        int n=arr.length;
        int result=sum(arr,n);
        System.out.println(result);
    }
    public static int sum(int arr[], int n){
        if (n == 0) {
            return 0;
        }
        return arr[n - 1] + sum(arr, n - 1);
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Hidden.java } ; i
{ java Hidden }
10
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

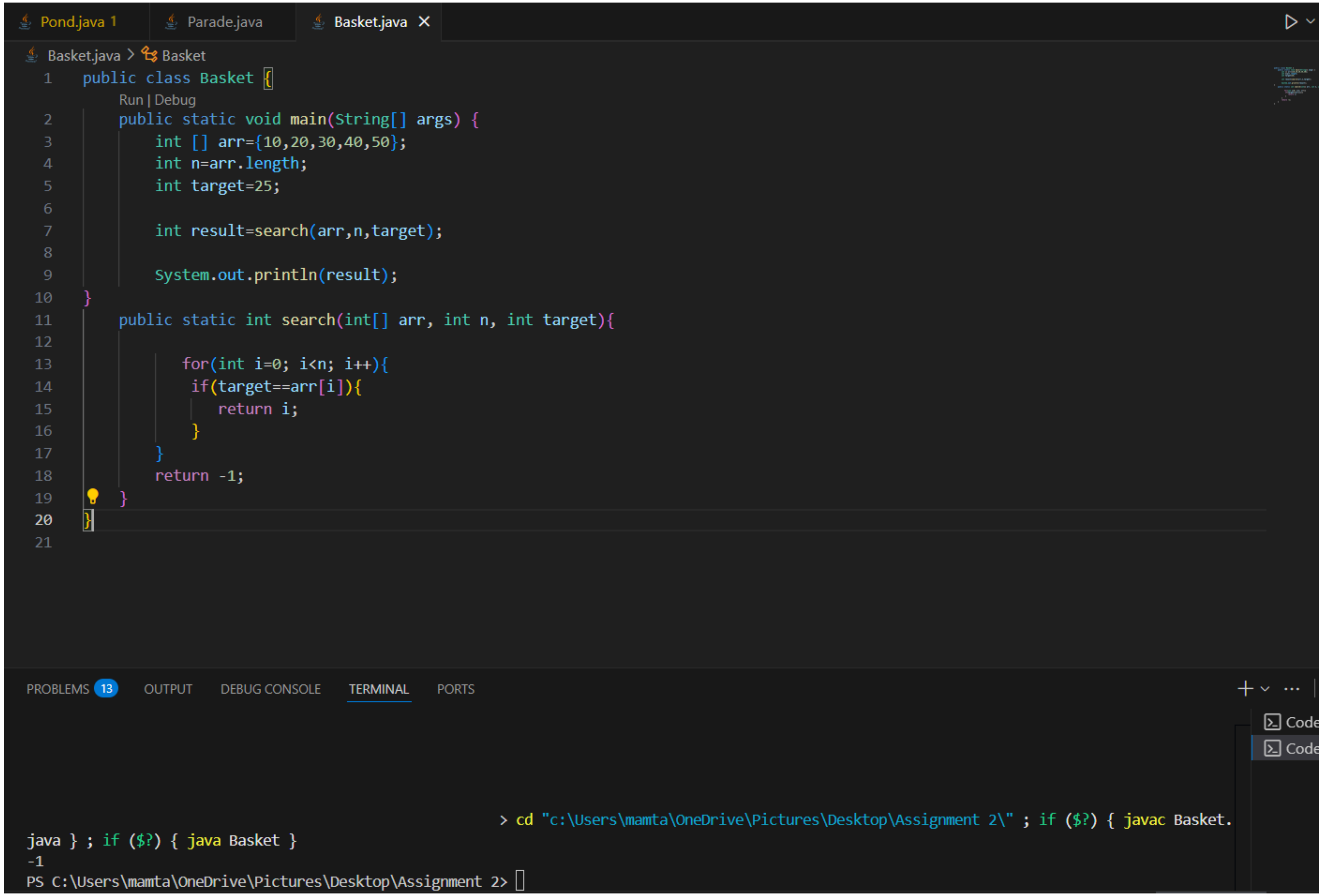
Q16. The Ancient Scroll



```java
public class AncientScroll {
    public static void main(String[] args){
        int [] arr={2,5,7,8};
        int n=arr.length;
        int target=7;
        int result=Search(arr,n,target);
        System.out.println(result);
    }
    public static int Search(int[] arr, int n, int target){

        for(int i=0; i<n; i++){
            if(target==arr[i]){
                return i;
            }
        }
        return -1;

    }
}

```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Ancient
Scroll.java } ; if ($?) { java AncientScroll }
2
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

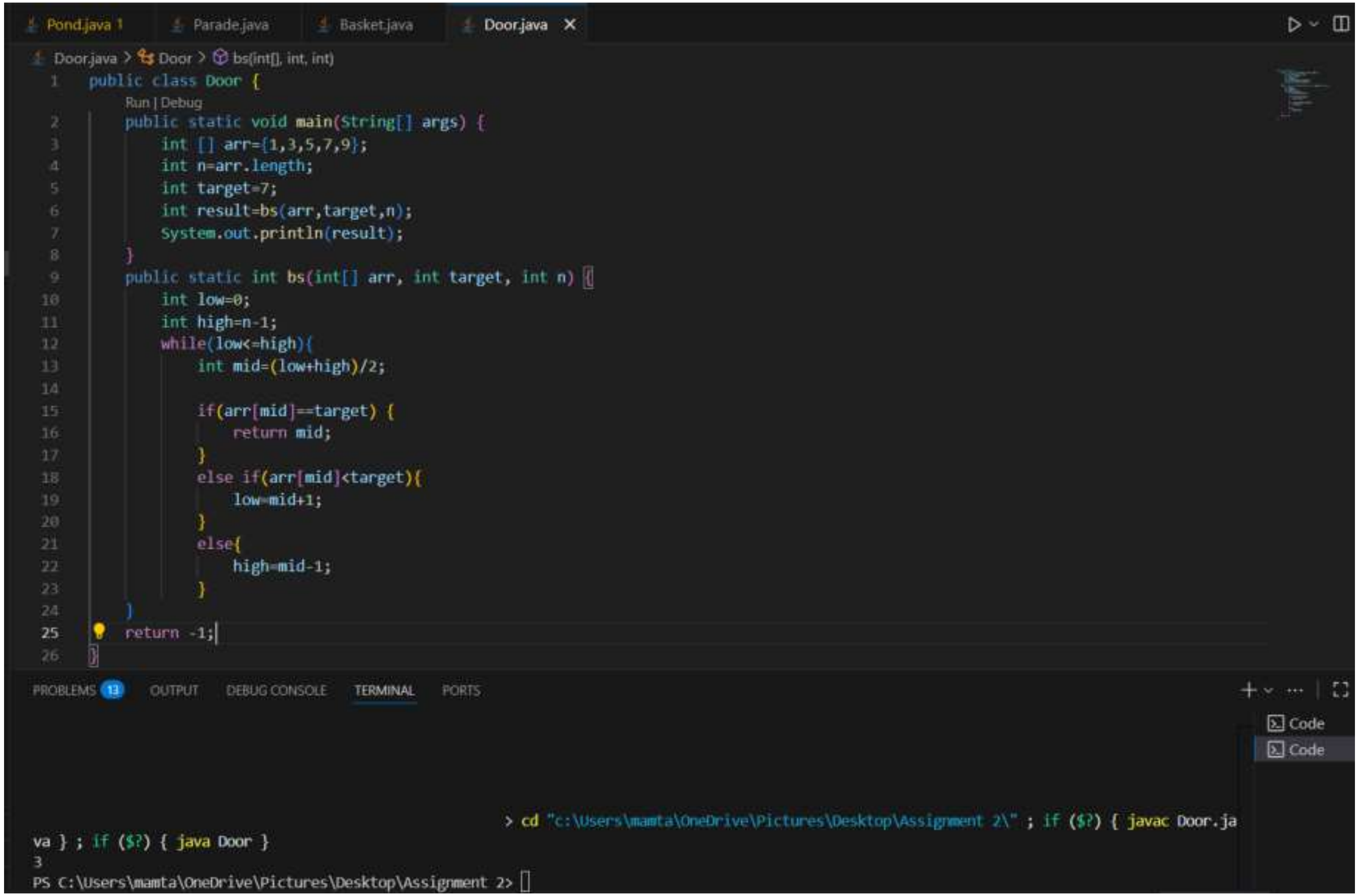
Q17. The Farmer’s Basket



```java
public class Basket {
    public static void main(String[] args) {
        int [] arr={10,20,30,40,50};
        int n=arr.length;
        int target=25;

        int result=search(arr,n,target);

        System.out.println(result);
}
    public static int search(int[] arr, int n, int target){

          for(int i=0; i<n; i++){
           if(target==arr[i]){
              return i;
           }
        }
        return -1;
    }
}

```

```
                                                 > cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Basket.
java } ; if ($?) { java Basket }
-1
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

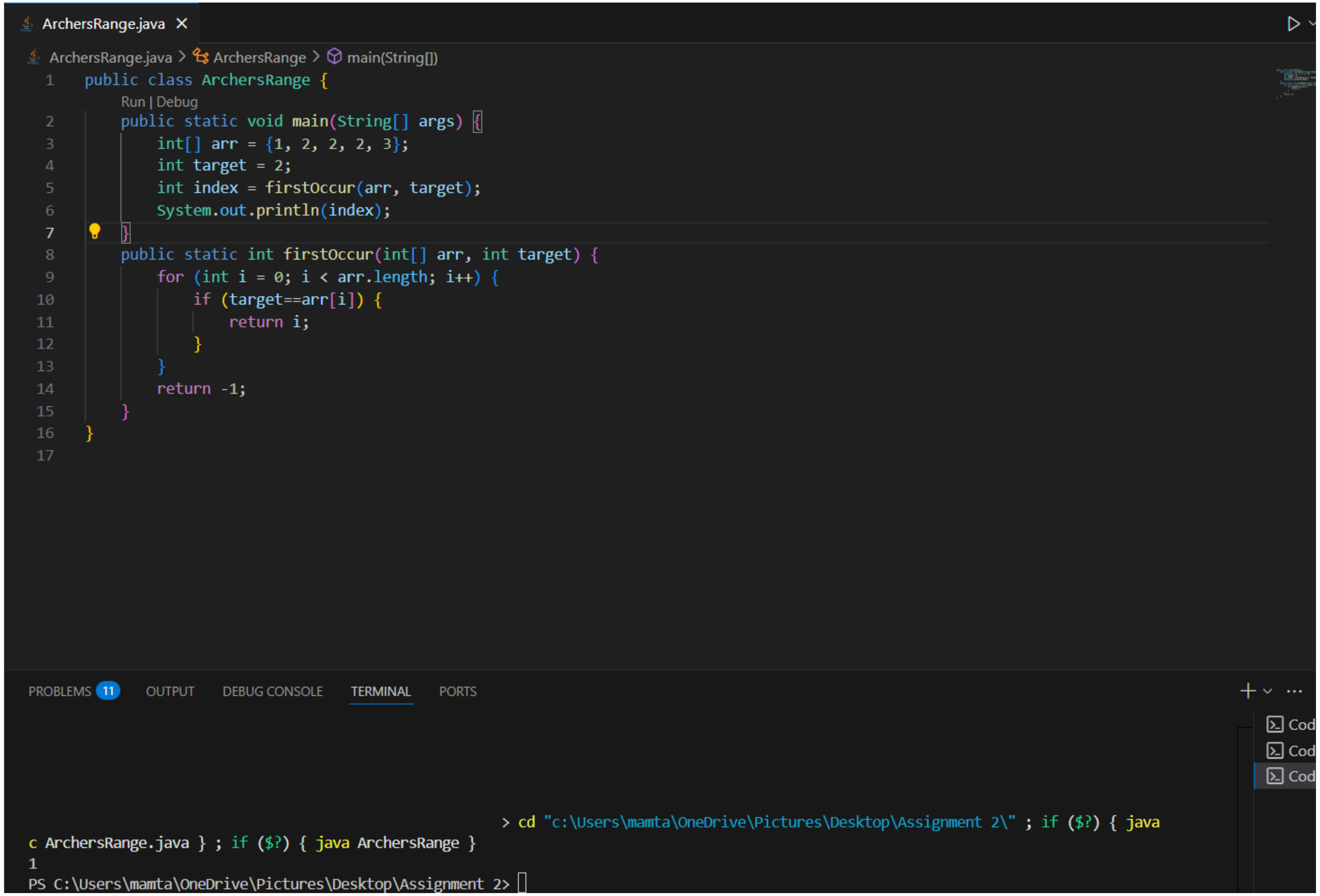
Q18. The Secret Door



```java
public class Door {
    public static void main(String[] args) {
        int [] arr={1,3,5,7,9};
        int n=arr.length;
        int target=7;
        int result=bs(arr,target,n);
        System.out.println(result);
    }
    public static int bs(int[] arr, int target, int n) {
        int low=0;
        int high=n-1;
        while(low<=high){
            int mid=(low+high)/2;

            if(arr[mid]==target) {
                return mid;
            }
            else if(arr[mid]<target){
                low=mid+1;
            }
            else{
                high=mid-1;
            }
        }
    return -1;
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Door.ja
va } ; if ($?) { java Door }
3
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

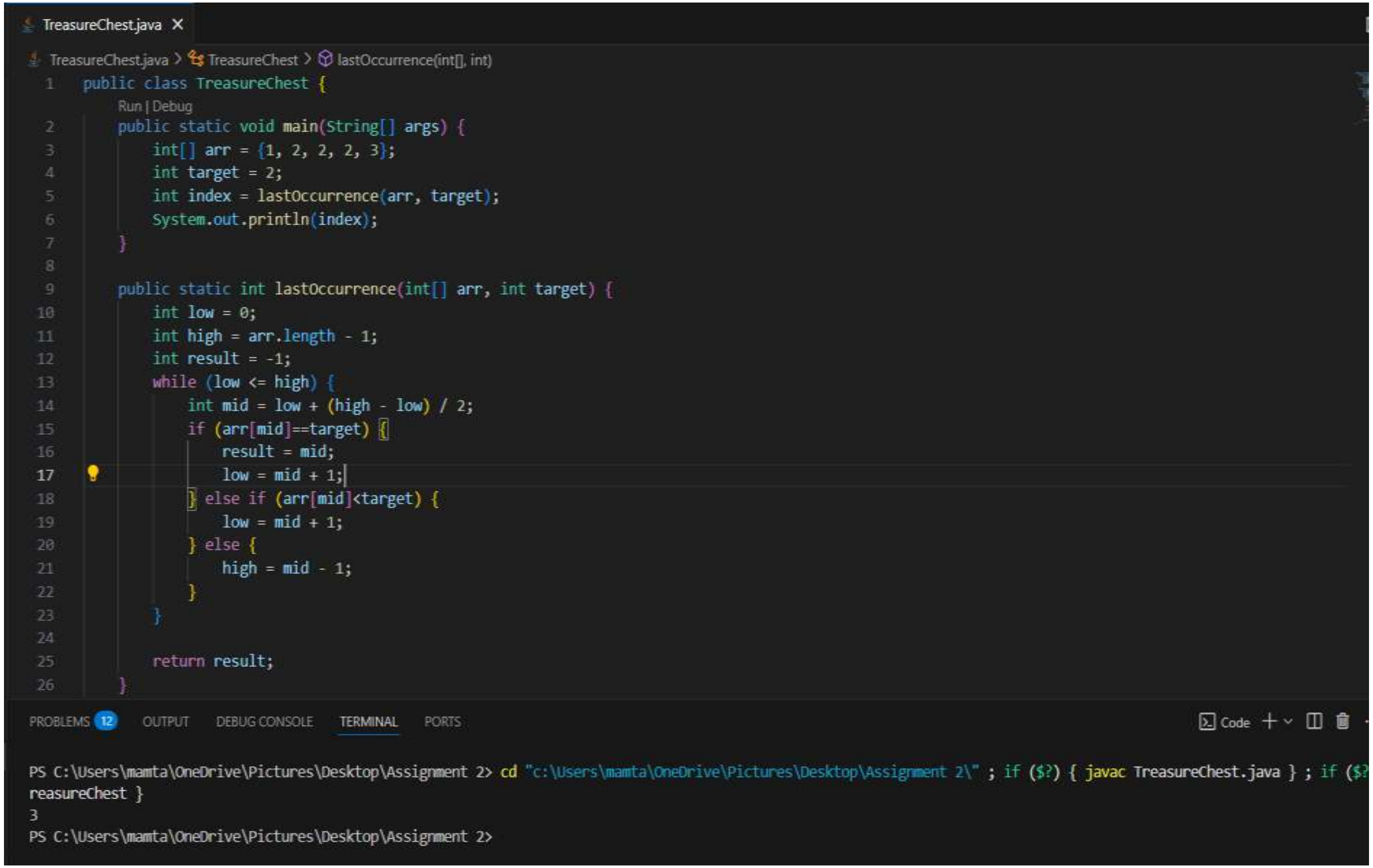
Q19. The Archer’s Range



```java
public class ArchersRange {
    public static void main(String[] args) {
        int[] arr = {1, 2, 2, 2, 3};
        int target = 2;
        int index = firstOccur(arr, target);
        System.out.println(index);
    }
    public static int firstOccur(int[] arr, int target) {
        for (int i = 0; i < arr.length; i++) {
            if (target==arr[i]) {
                return i;
            }
        }
        return -1;
    }
}
```

```
> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { java
c ArchersRange.java } ; if ($?) { java ArchersRange }
1
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q20. The Treasure Chest

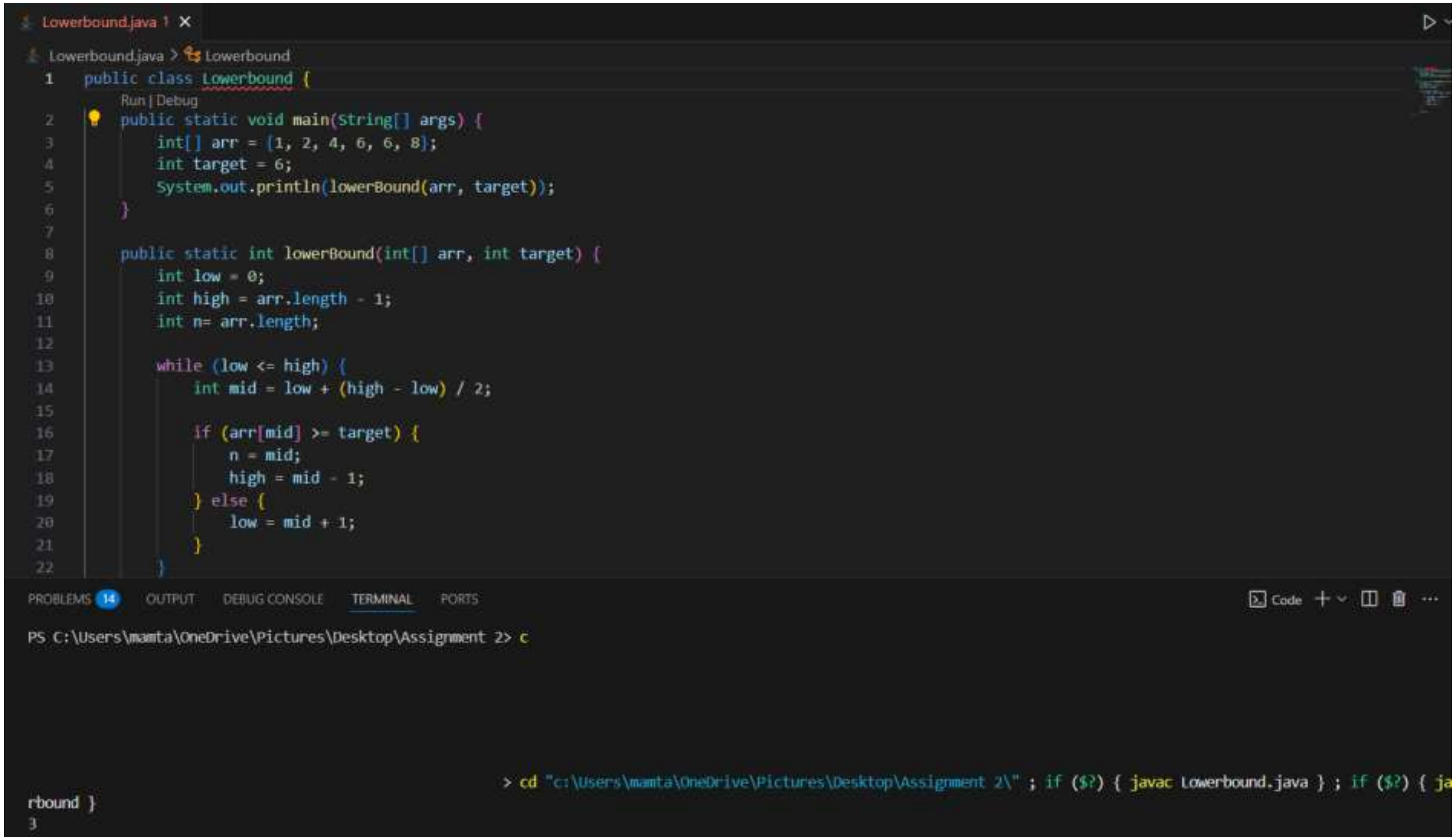


TreasureChest.java
TreasureChest.java > TreasureChest > lastOccurrence(int[], int)
Run | Debug
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code

```java
public class TreasureChest {
    public static void main(String[] args) {
        int[] arr = {1, 2, 2, 2, 3};
        int target = 2;
        int index = lastOccurrence(arr, target);
        System.out.println(index);
    }

    public static int lastOccurrence(int[] arr, int target) {
        int low = 0;
        int high = arr.length - 1;
        int result = -1;
        while (low <= high) {
            int mid = low + (high - low) / 2;
            if (arr[mid]==target) {
                result = mid;
                low = mid + 1;
            } else if (arr[mid]<target) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }

        return result;
    }
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac TreasureChest.java } ; if ($?
reasureChest }
3
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

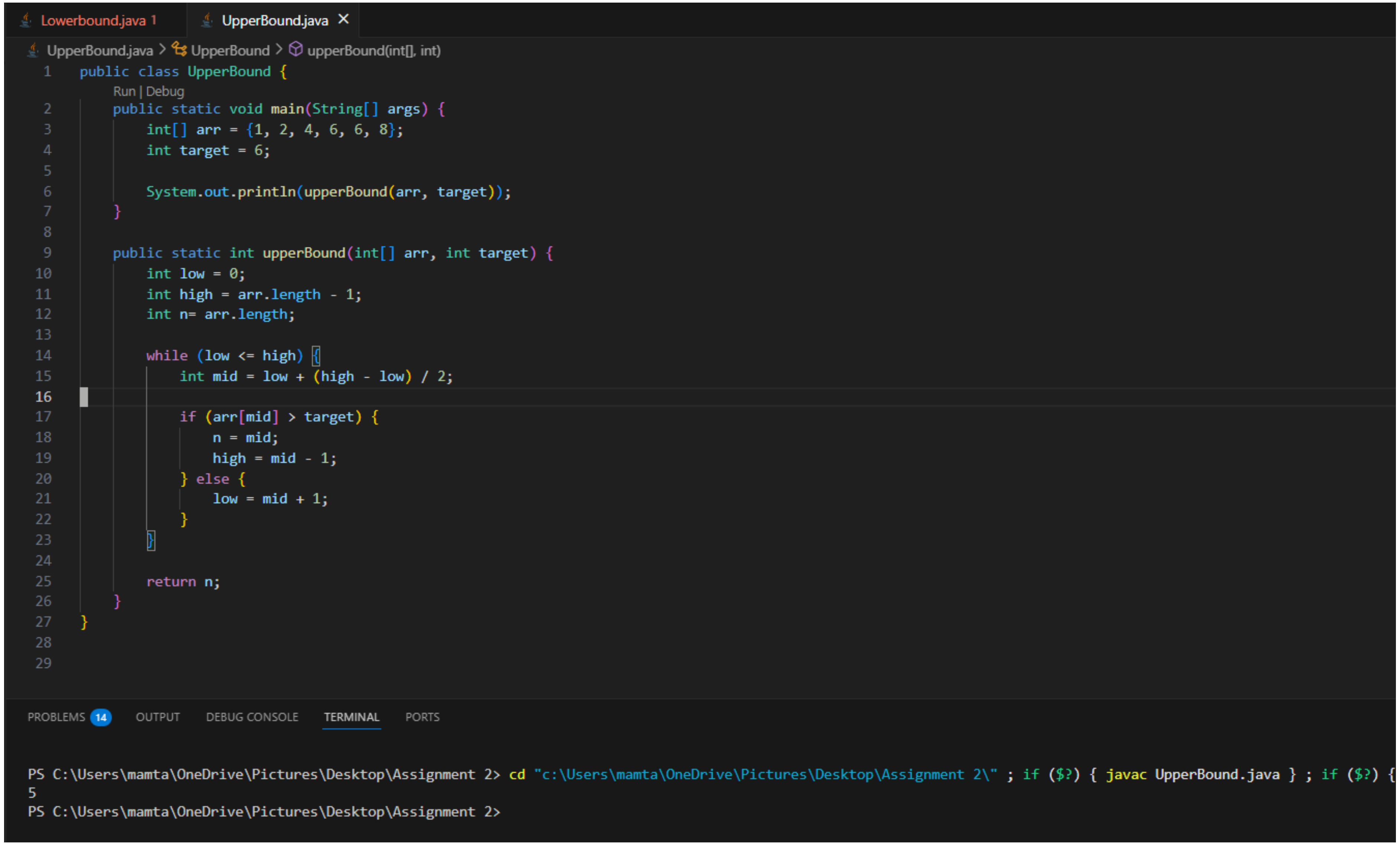
Q21.The first index where the element is greater than or equal to the target.



```java
public class Lowerbound {
    public static void main(String[] args) {
        int[] arr = {1, 2, 4, 6, 6, 8};
        int target = 6;
        System.out.println(lowerBound(arr, target));
    }

    public static int lowerBound(int[] arr, int target) {
        int low = 0;
        int high = arr.length - 1;
        int n= arr.length;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] >= target) {
                n = mid;
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> c
                                        > cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Lowerbound.java } ; if ($?) { ja
rbound }
3
```

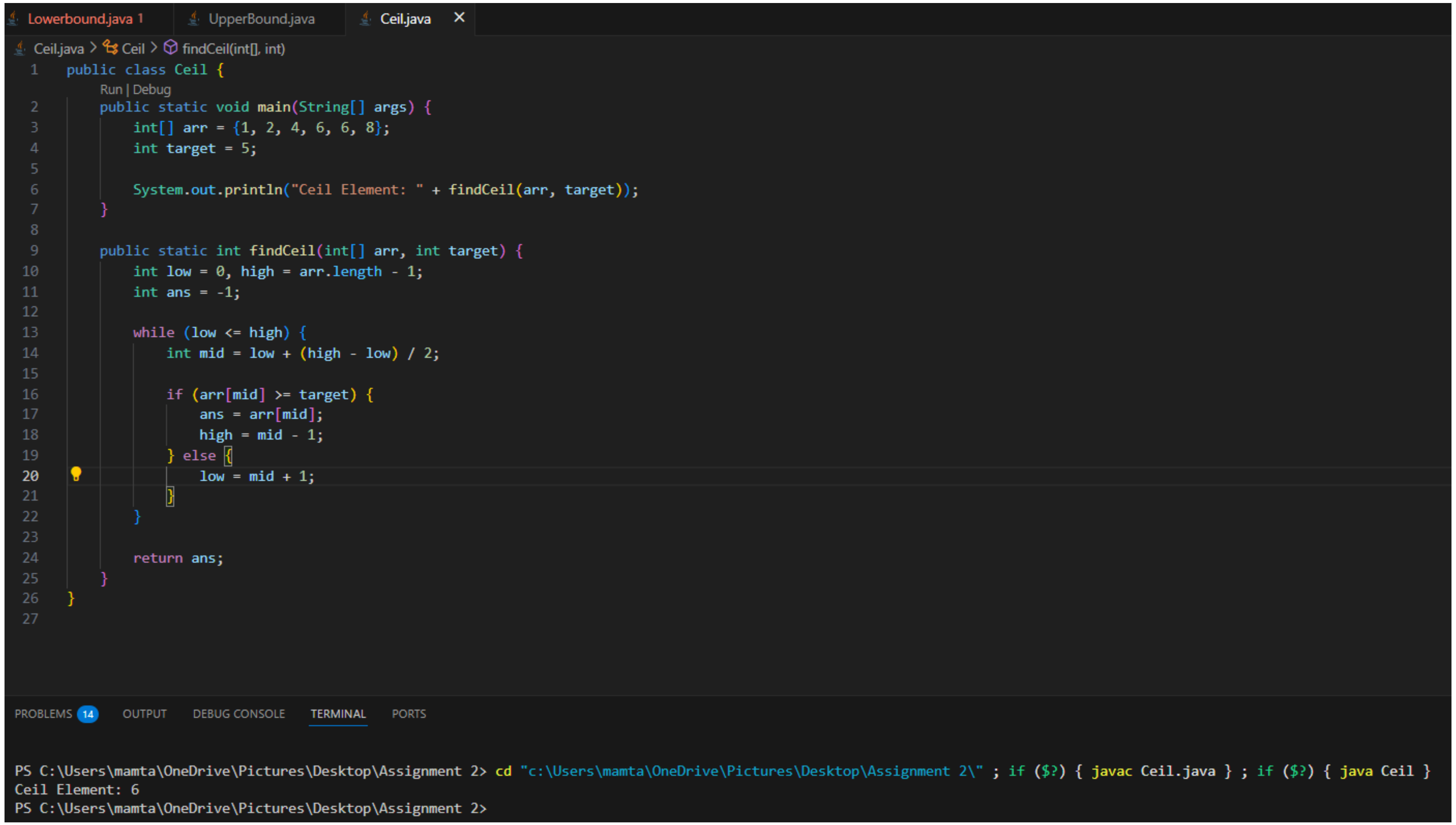
Q22.The first index where the element is strictly greater than the target.



```java
public class UpperBound {
    public static void main(String[] args) {
        int[] arr = {1, 2, 4, 6, 6, 8};
        int target = 6;

        System.out.println(upperBound(arr, target));
    }

    public static int upperBound(int[] arr, int target) {
        int low = 0;
        int high = arr.length - 1;
        int n= arr.length;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] > target) {
                n = mid;
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }

        return n;
    }
}
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac UpperBound.java } ; if ($?) {
5
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

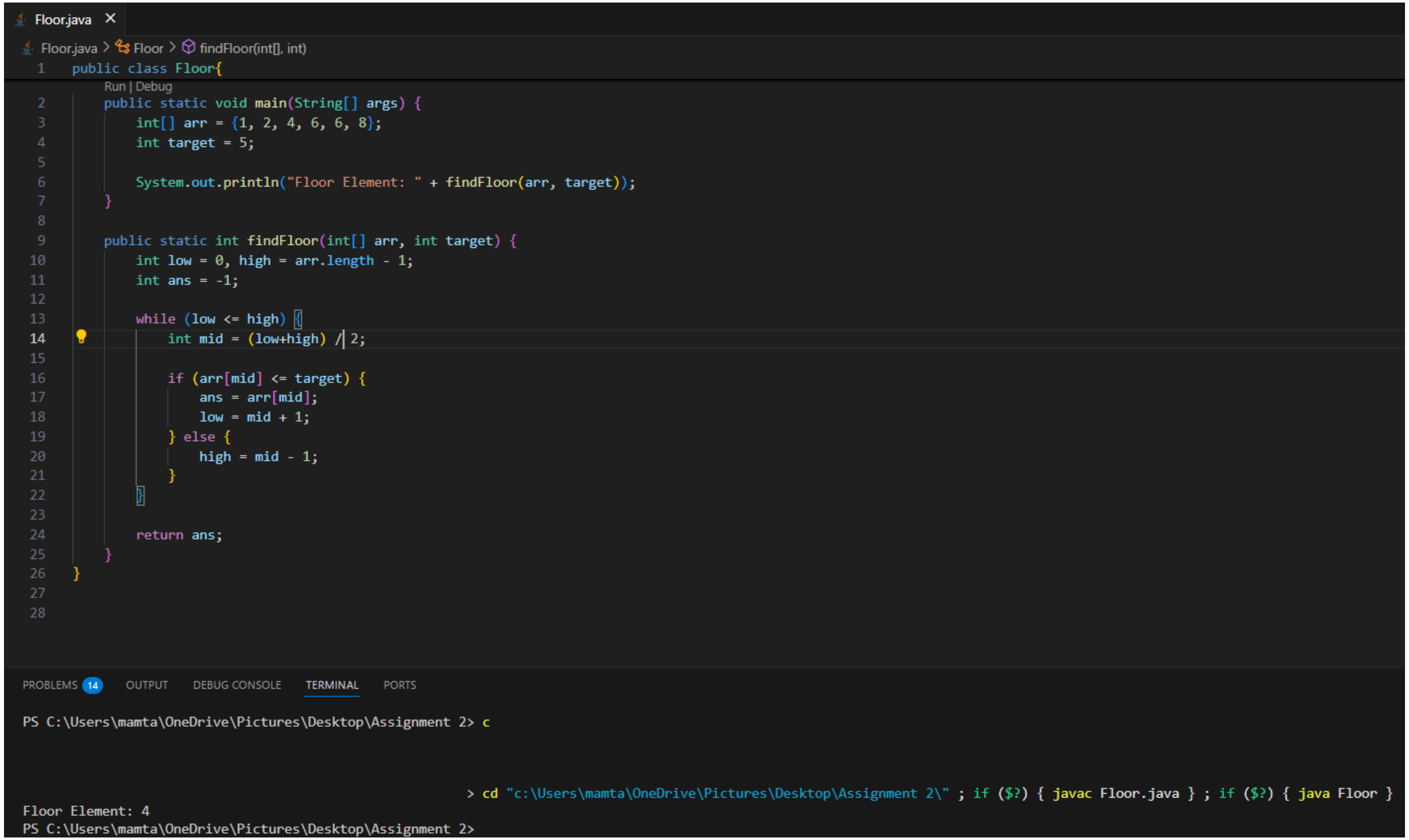
Q23.The smallest element ≥ target (actual value, not index).



```java
public class Ceil {
    public static void main(String[] args) {
        int[] arr = {1, 2, 4, 6, 6, 8};
        int target = 5;

        System.out.println("Ceil Element: " + findCeil(arr, target));
    }

    public static int findCeil(int[] arr, int target) {
        int low = 0, high = arr.length - 1;
        int ans = -1;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] >= target) {
                ans = arr[mid];
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }

        return ans;
    }
}
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Ceil.java } ; if ($?) { java Ceil }
Ceil Element: 6
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

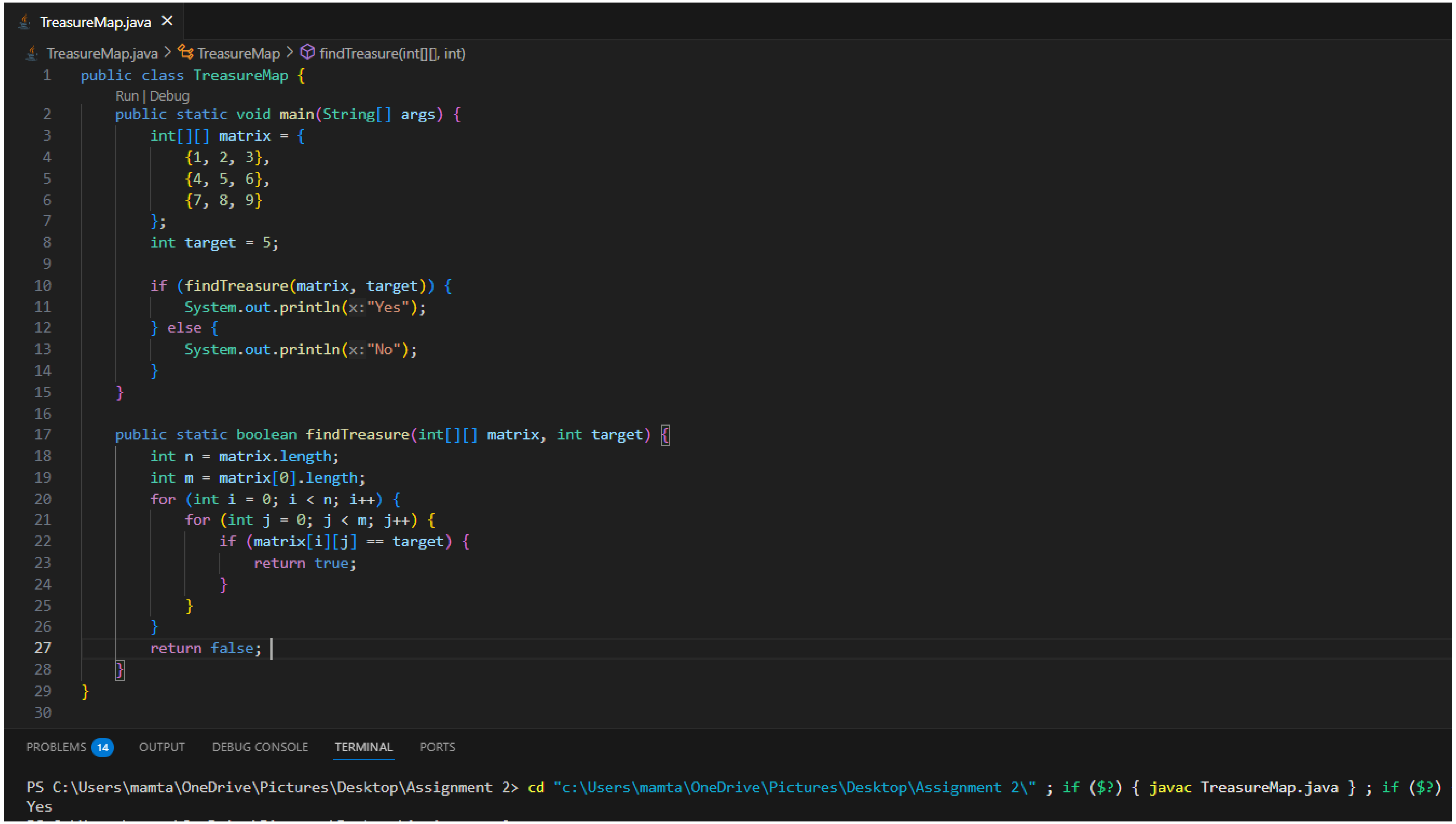
Q24. The largest element ≤ target.



```java
public class Floor{
    public static void main(String[] args) {
        int[] arr = {1, 2, 4, 6, 6, 8};
        int target = 5;

        System.out.println("Floor Element: " + findFloor(arr, target));
    }

    public static int findFloor(int[] arr, int target) {
        int low = 0, high = arr.length - 1;
        int ans = -1;

        while (low <= high) {
            int mid = (low+high) / 2;

            if (arr[mid] <= target) {
                ans = arr[mid];
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }

        return ans;
    }
}
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> c


                                        > cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Floor.java } ; if ($?) { java Floor }
Floor Element: 4
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q25. The Treasure Map (Linear Search)



TreasureMap.java ×

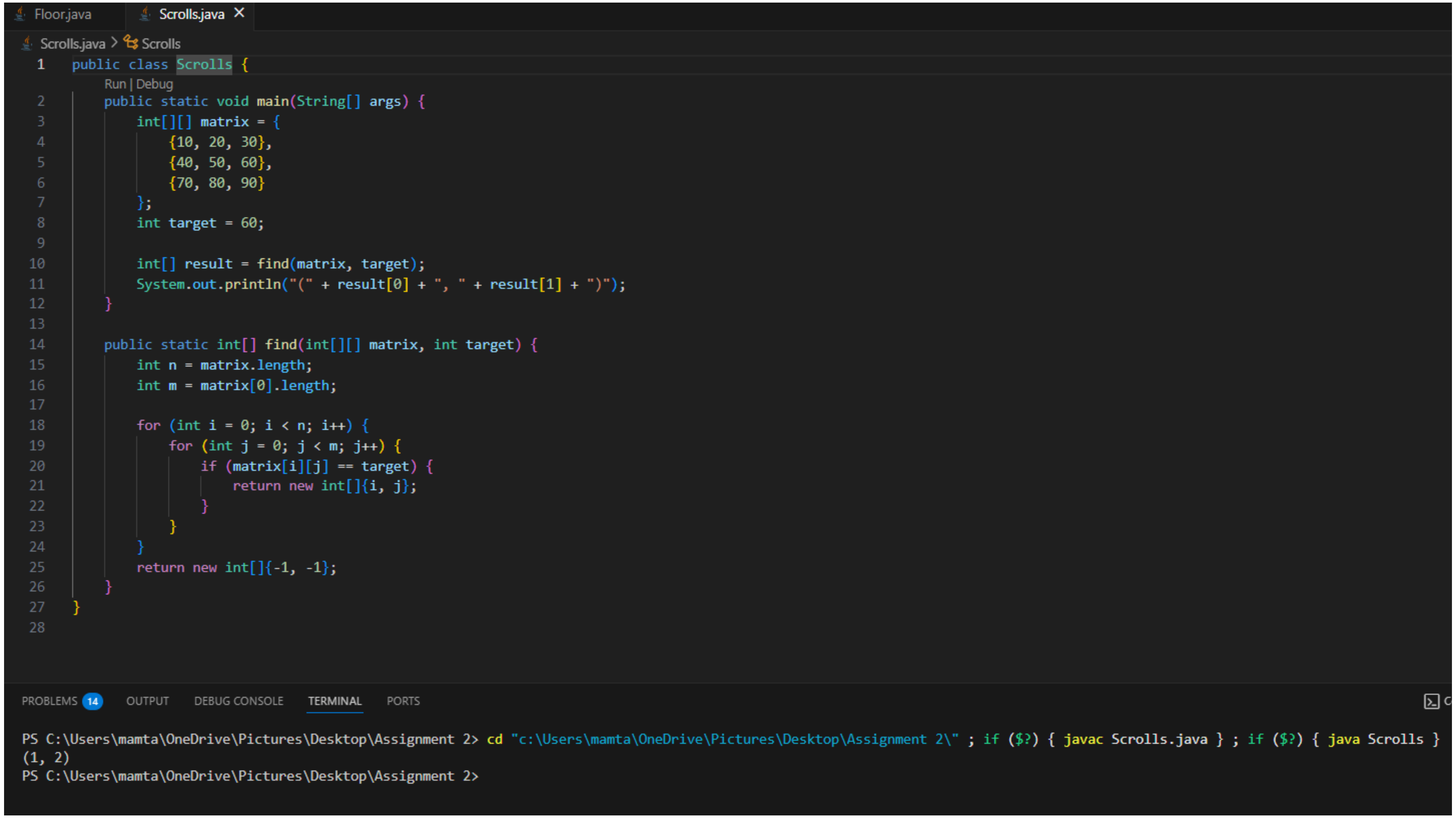
TreasureMap.java > TreasureMap > findTreasure(int[][], int)

```java
public class TreasureMap {
    Run | Debug
    public static void main(String[] args) {
        int[][] matrix = {
            {1, 2, 3},
            {4, 5, 6},
            {7, 8, 9}
        };
        int target = 5;

        if (findTreasure(matrix, target)) {
            System.out.println(x:"Yes");
        } else {
            System.out.println(x:"No");
        }
    }

    public static boolean findTreasure(int[][] matrix, int target) {
        int n = matrix.length;
        int m = matrix[0].length;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                if (matrix[i][j] == target) {
                    return true;
                }
            }
        }
        return false;
    }
}
```

PROBLEMS 14　OUTPUT　DEBUG CONSOLE　TERMINAL　PORTS

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac TreasureMap.java } ; if ($?)
Yes
```

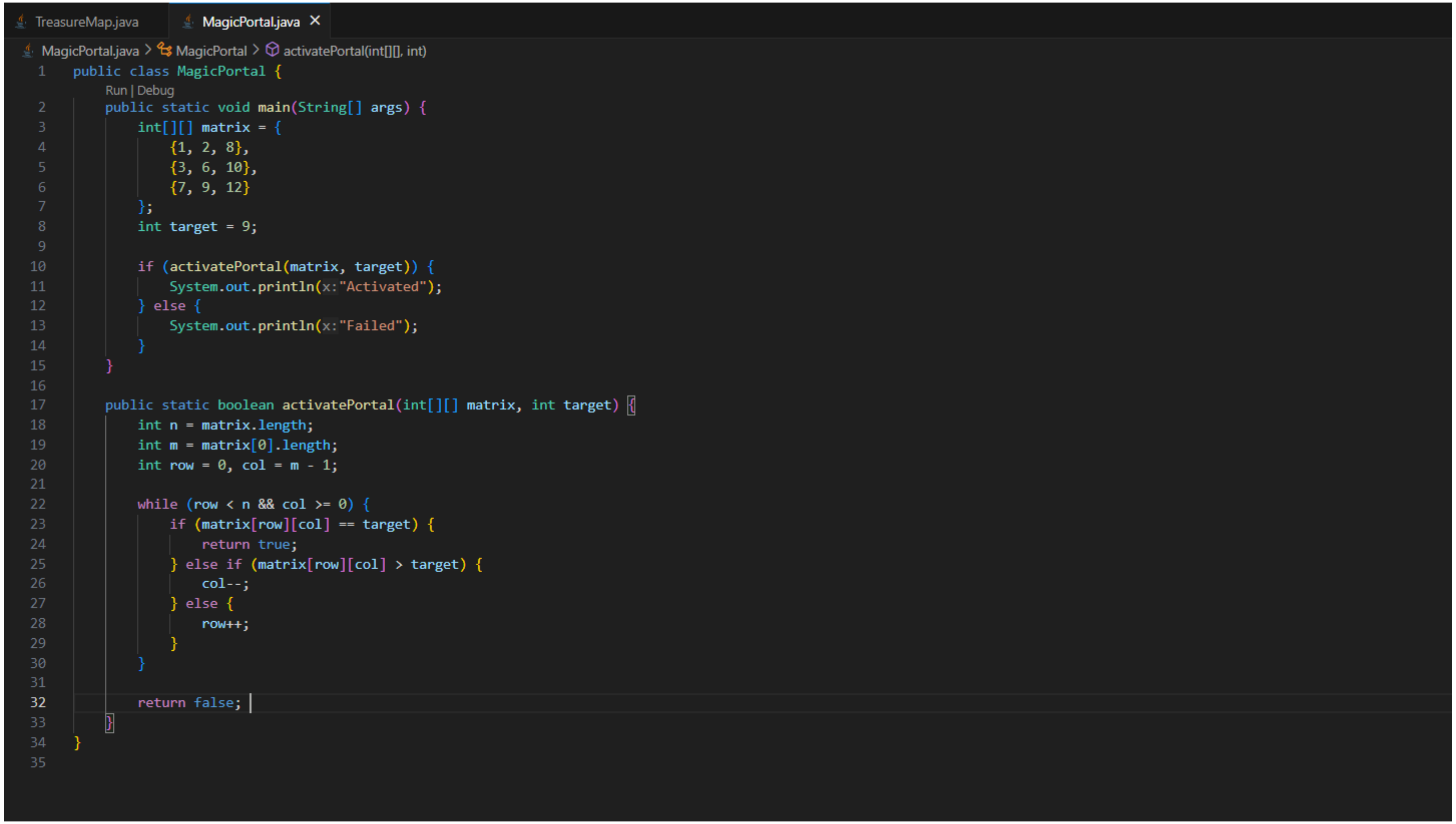
Q26. The Magical Scrolls (Linear Search Return Index)



```java
public class Scrolls {
    public static void main(String[] args) {
        int[][] matrix = {
            {10, 20, 30},
            {40, 50, 60},
            {70, 80, 90}
        };
        int target = 60;

        int[] result = find(matrix, target);
        System.out.println("(" + result[0] + ", " + result[1] + ")");
    }

    public static int[] find(int[][] matrix, int target) {
        int n = matrix.length;
        int m = matrix[0].length;

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                if (matrix[i][j] == target) {
                    return new int[]{i, j};
                }
            }
        }
        return new int[]{-1, -1};
    }
}
```

```
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2> cd "c:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2\" ; if ($?) { javac Scrolls.java } ; if ($?) { java Scrolls }
(1, 2)
PS C:\Users\mamta\OneDrive\Pictures\Desktop\Assignment 2>
```

Q30. The Magic Portal (Binary Search 2D)


TreasureMap.java
MagicPortal.java
MagicPortal.java > MagicPortal > activatePortal(int[][], int)
Run | Debug

```java
public class MagicPortal {
    public static void main(String[] args) {
        int[][] matrix = {
            {1, 2, 8},
            {3, 6, 10},
            {7, 9, 12}
        };
        int target = 9;

        if (activatePortal(matrix, target)) {
            System.out.println(x:"Activated");
        } else {
            System.out.println(x:"Failed");
        }
    }

    public static boolean activatePortal(int[][] matrix, int target) {
        int n = matrix.length;
        int m = matrix[0].length;
        int row = 0, col = m - 1;

        while (row < n && col >= 0) {
            if (matrix[row][col] == target) {
                return true;
            } else if (matrix[row][col] > target) {
                col--;
            } else {
                row++;
            }
        }

        return false;
    }
}
```