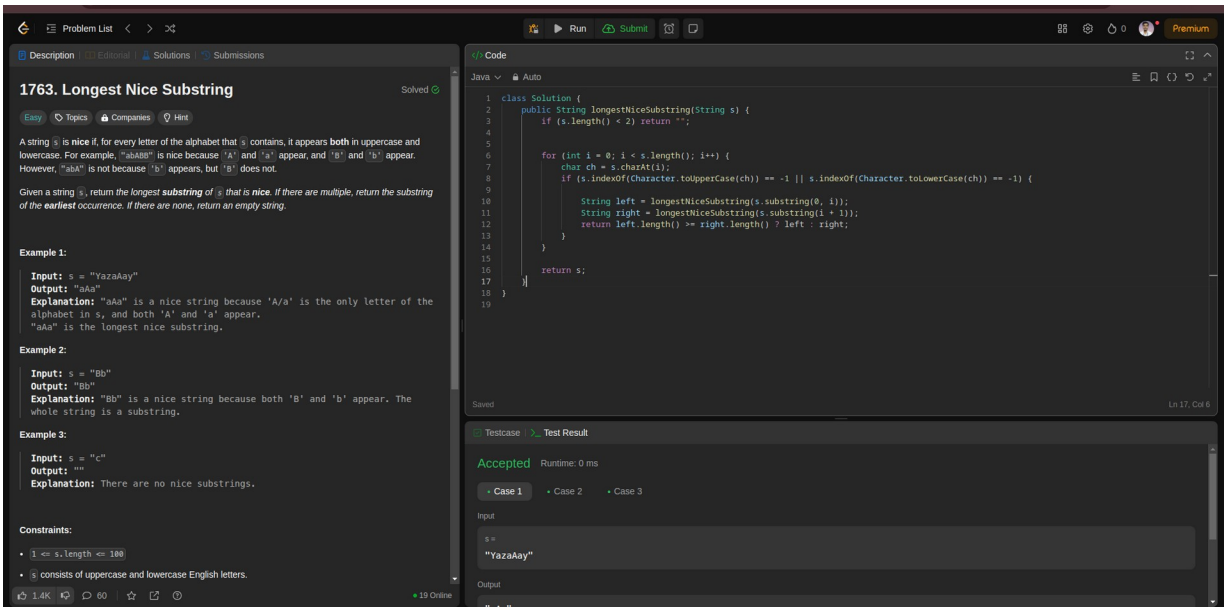
Assignment 4

Student Name: Kumar Devashish		**UID**	**:**22BCS10248
Branch	**:** BE-CSE	**Sect./Grp**	**:**FL_IOT-602-A
Semester	**:** 6th	**Date of Performance**	**:**21/02/25
Subject Name	**:** AP Lab	**Subject Code**	**:**22CSP-351

1. Aim: 1763. Longest Nice Substring

```
class Solution {
    public String longestNiceSubstring(String s) {
        if (s.length() < 2) return "";



        for (int i = 0; i < s.length(); i++) {
            char ch = s.charAt(i);
            if (s.indexOf(Character.toUpperCase(ch)) == -1 ||
s.indexOf(Character.toLowerCase(ch)) == -1) {

                String left = longestNiceSubstring(s.substring(0, i));
                String right = longestNiceSubstring(s.substring(i + 1));
                return left.length() >= right.length() ? left : right;
            }
        }

        return s;
    }
}
```

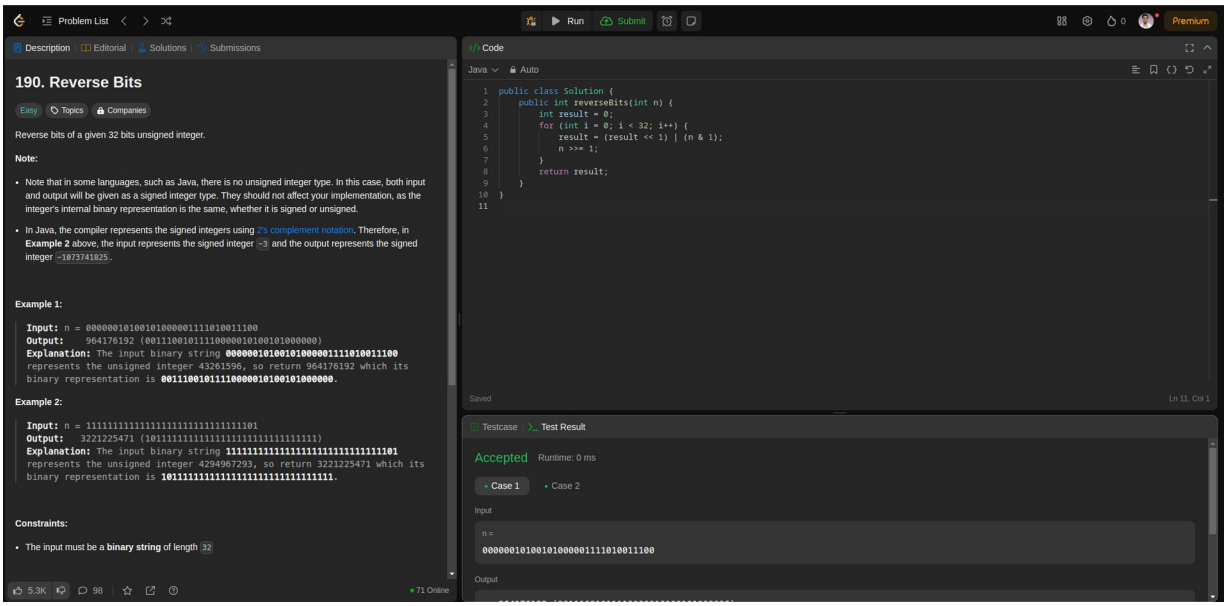
Output:



2. Aim: 190. Reverse Bits

```
public class Solution {
    public int reverseBits(int n) {
        int result = 0;
        for (int i = 0; i < 32; i++) {
            result = (result << 1) | (n & 1);
            n >>= 1;
        }
        return result;
    }
}
```

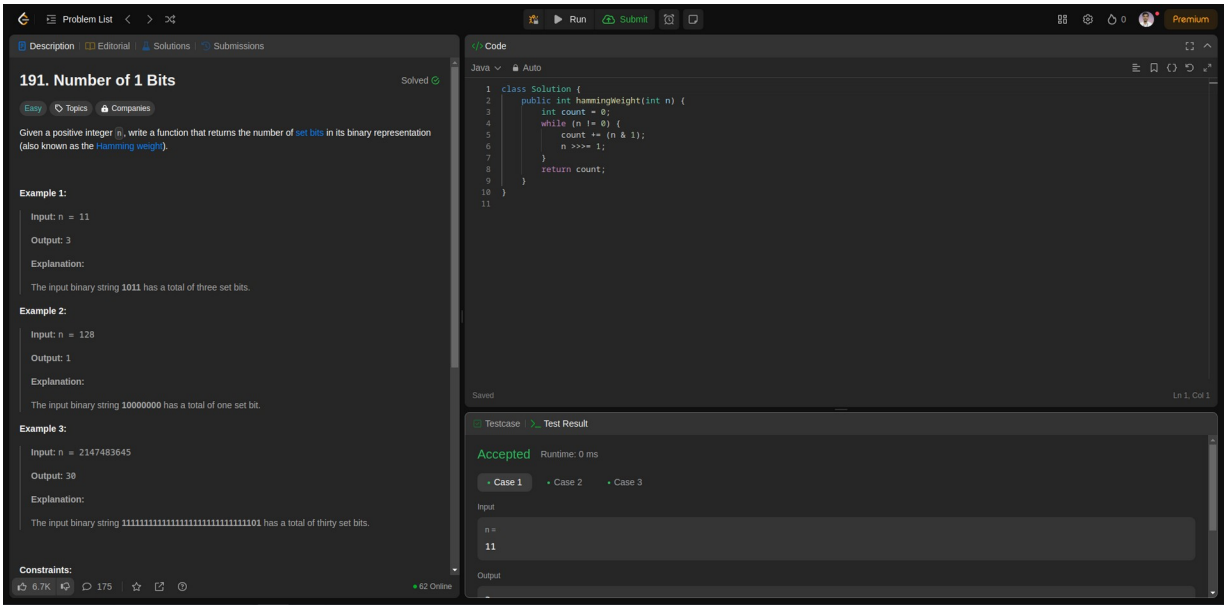
Output:



3. Aim: 191. Number of 1 Bits

```
class Solution {
    public int hammingWeight(int n) {
        int count = 0;
        while (n != 0) {
            count += (n & 1);
            n >>>= 1;
        }
        return count;
    }
}
```

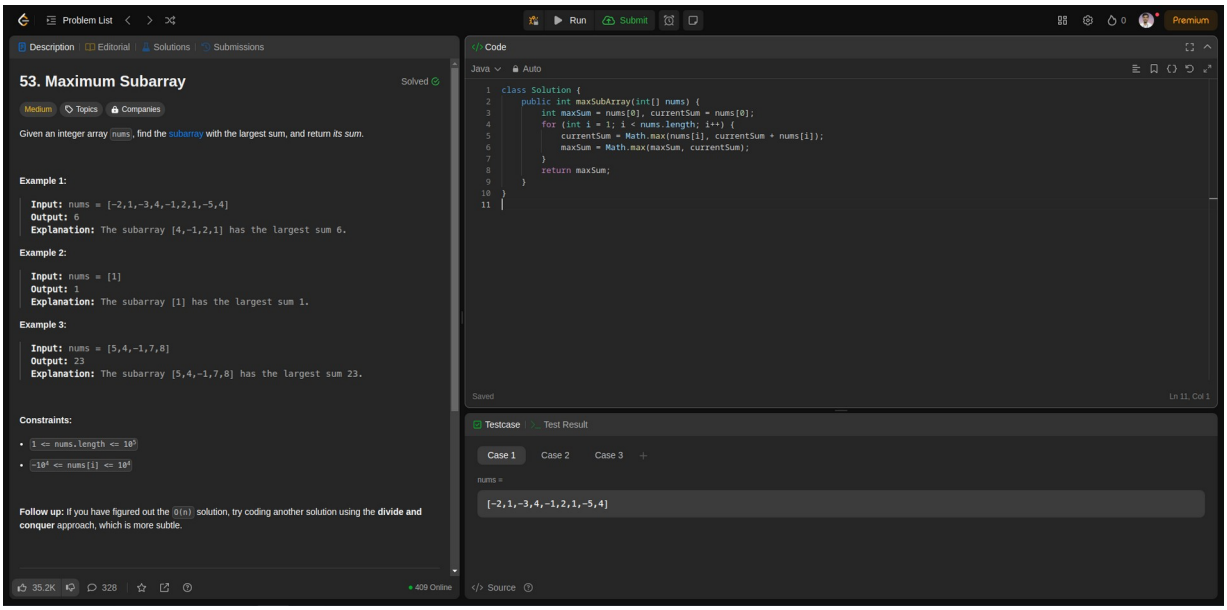
Output:



4. Aim: 53. Maximum Subarray

```
class Solution {
    public int maxSubArray(int[] nums) {
        int maxSum = nums[0], currentSum = nums[0];
        for (int i = 1; i < nums.length; i++) {
            currentSum = Math.max(nums[i], currentSum + nums[i]);
            maxSum = Math.max(maxSum, currentSum);
        }
        return maxSum;
    }
}
```

Output:

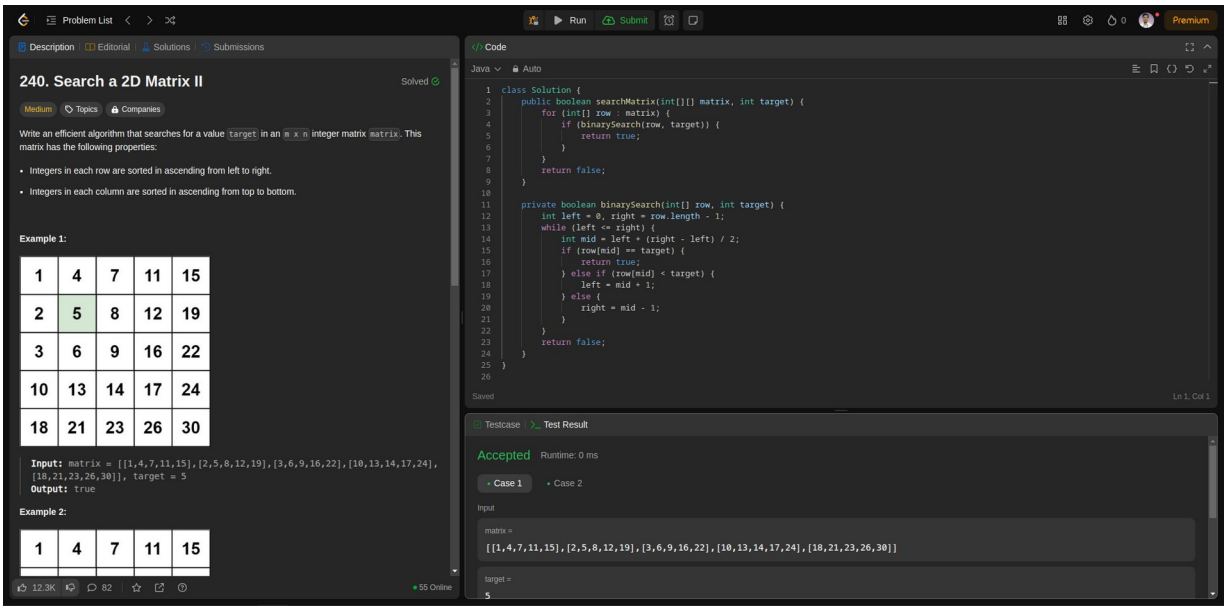


5. Aim: 240. Search a 2D Matrix II

Implementation/ Code:

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = 0, col = matrix[0].size() - 1;
        while (row < matrix.size() && col >= 0) {
            if (matrix[row][col] == target) return true;
            matrix[row][col] > target ? col-- : row++;
        }
        return false;
    }
};
```

Output:



6. Aim: 372. Super Pow

```
class Solution {
    private static final int MOD = 1337;

    public int superPow(int a, int[] b) {
        a %= MOD;
        int result = 1;
        for (int digit : b) {
            result = power(result, 10) * power(a, digit) % MOD;
        }
        return result;
    }
```

```
    private int power(int base, int exp) {
        int res = 1;
        while (exp > 0) {
            if ((exp & 1) == 1) res = res * base % MOD;
            base = base * base % MOD;
            exp >>= 1;
        }
        return res;
    }
}
```

Output: