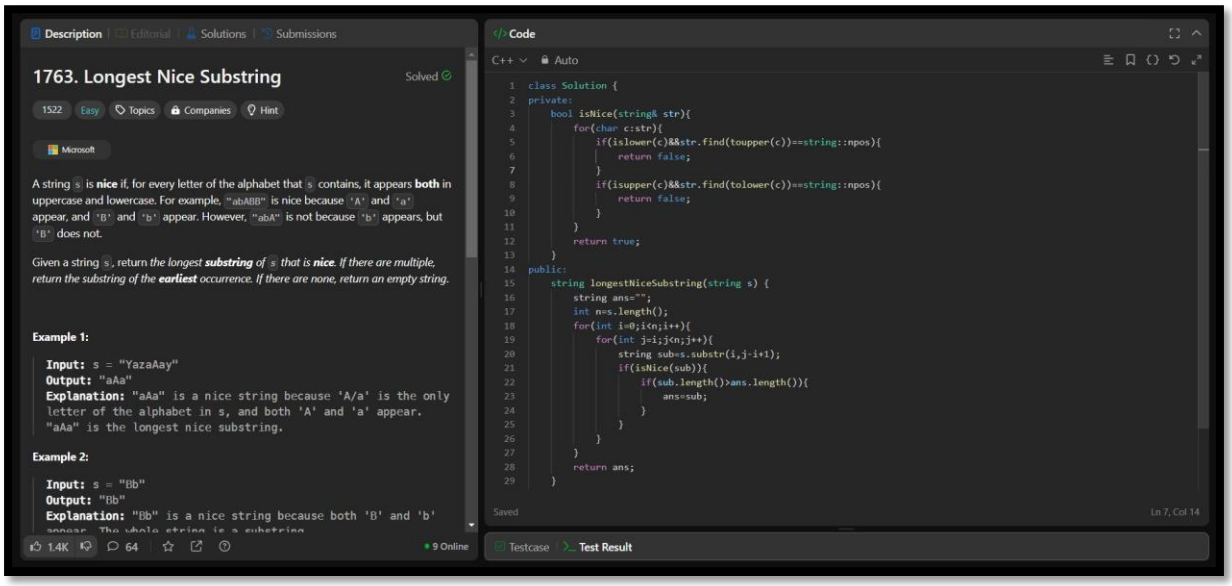
Name: - Navneet Sharma

UID: - 22BCS15680

Sec/Group: - IOT_609/B

AP worksheet

1. Longest Nice Substring:



1763. Longest Nice Substring — Solved

1522 | Easy | Topics | Companies | Hint

Microsoft

A string `s` is **nice** if, for every letter of the alphabet that `s` contains, it appears **both** in uppercase and lowercase. For example, `"abABB"` is nice because `'A'` and `'a'` appear, and `'B'` and `'b'` appear. However, `"abA"` is not because `'b'` appears, but `'B'` does not.

Given a string `s`, return *the longest **substring** of `s` that is **nice**. If there are multiple, return the substring of the **earliest** occurrence. If there are none, return an empty string.*

Example 1:

```
Input: s = "YazaAay"
Output: "aAa"
Explanation: "aAa" is a nice string because 'A/a' is the only
letter of the alphabet in s, and both 'A' and 'a' appear.
"aAa" is the longest nice substring.
```

Example 2:

```
Input: s = "Bb"
Output: "Bb"
Explanation: "Bb" is a nice string because both 'B' and 'b'
```

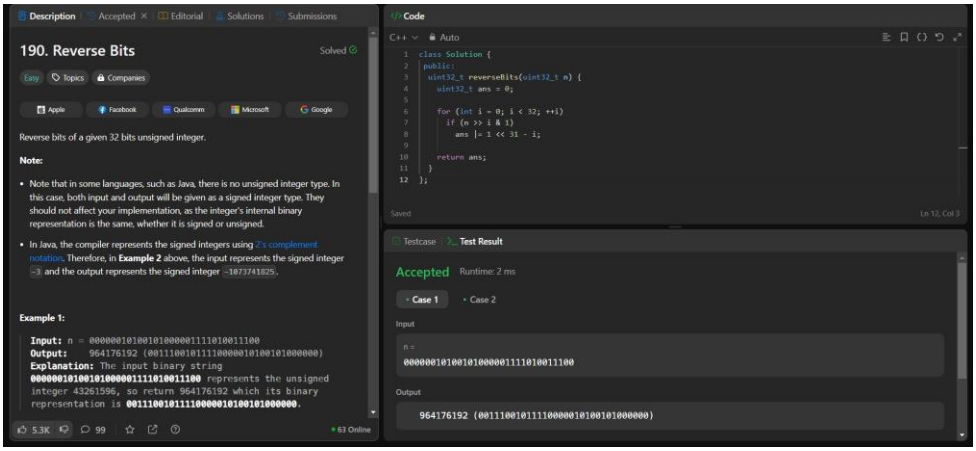
1.4K | 64 | 9 Online

```cpp
class Solution {
private:
    bool isNice(string& str){
        for(char c:str){
            if(islower(c)&&str.find(toupper(c))==string::npos){
                return false;
            }
            if(isupper(c)&&str.find(tolower(c))==string::npos){
                return false;
            }
        }
        return true;
    }
public:
    string longestNiceSubstring(string s) {
        string ans="";
        int n=s.length();
        for(int i=0;i<n;i++){
            for(int j=i;j<n;j++){
                string sub=s.substr(i,j-i+1);
                if(isNice(sub)){
                    if(sub.length()>ans.length()){
                        ans=sub;
                    }
                }
            }
        }
        return ans;
    }
```

Saved | Ln 7, Col 14

Testcase | Test Result

2. Reverse Bits



190. Reverse Bits — Solved

Easy | Topics | Companies

Apple | Facebook | Qualcomm | Microsoft | Google

Reverse bits of a given 32 bits unsigned integer.

Note:

- Note that in some languages, such as Java, there is no unsigned integer type. In this case, both input and output will be given as a signed integer type. They should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
- In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in **Example 2** above, the input represents the signed integer `-3` and the output represents the signed integer `-1073741825`.

Example 1:

```
Input: n = 00000010100101000001111010011100
Output:    964176192 (00111001011111000001010010100000)
Explanation: The input binary string
00000010100101000001111010011100 represents the unsigned
integer 43261596, so return 964176192 which its binary
representation is 00111001011111000001010010100000.
```

5.3K | 99 | 63 Online

```cpp
class Solution {
public:
  uint32_t reverseBits(uint32_t n) {
    uint32_t ans = 0;

    for (int i = 0; i < 32; ++i)
      if (n >> i & 1)
        ans |= 1 << 31 - i;

    return ans;
  }
};
```

Saved | Ln 12, Col 3

Testcase | Test Result

Accepted Runtime: 2 ms

Case 1 | Case 2

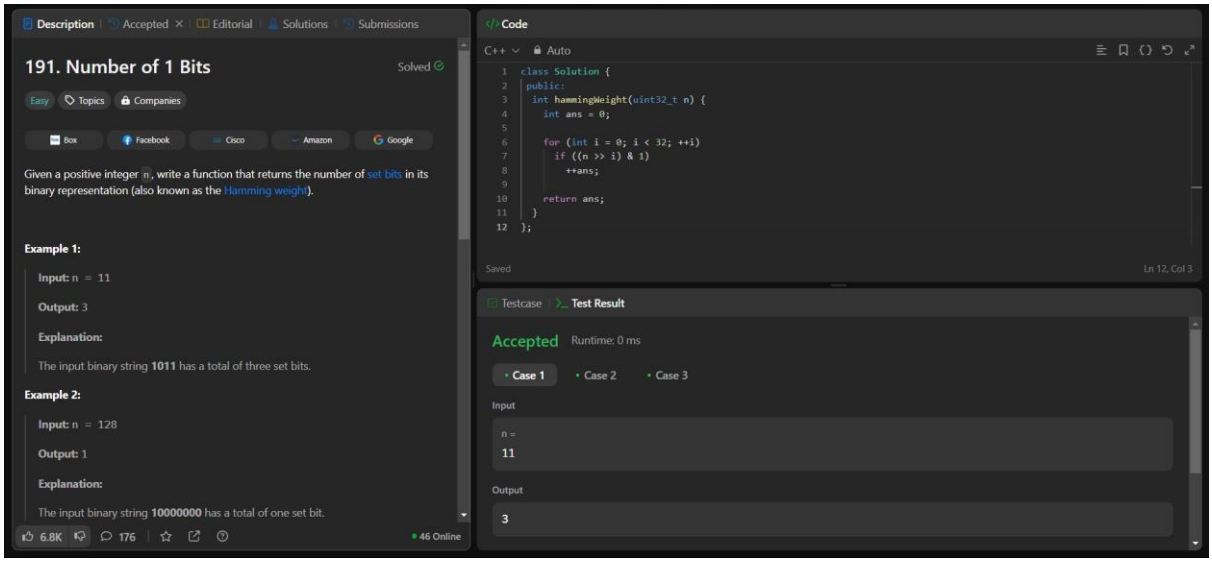
Input

```
n =
00000010100101000001111010011100
```

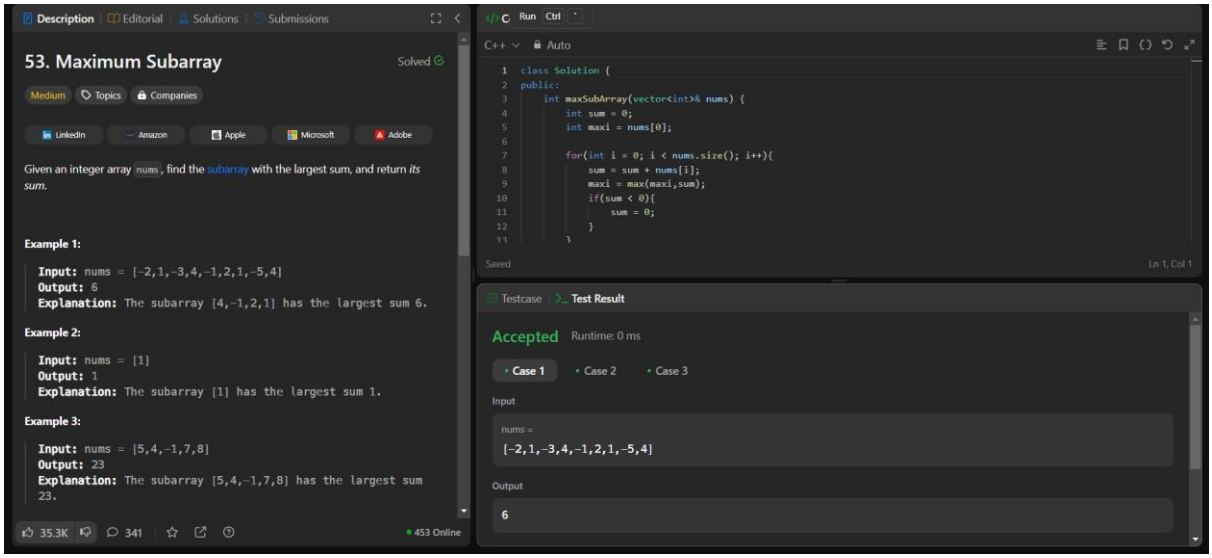
Output

```
964176192 (00111001011111000001010010100000)
```

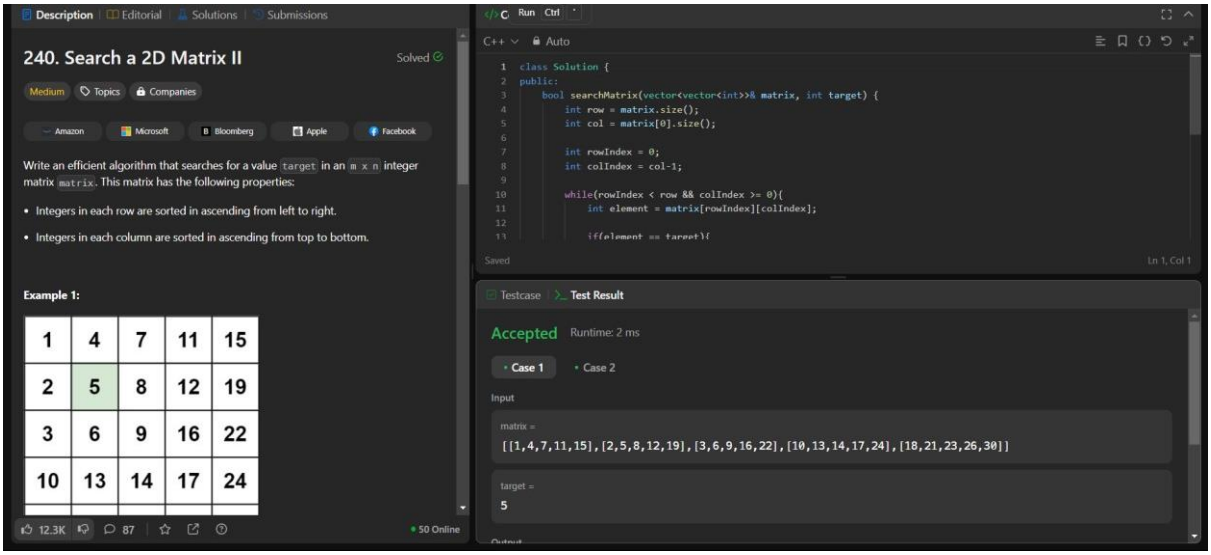
3. Number of 1 Bits:



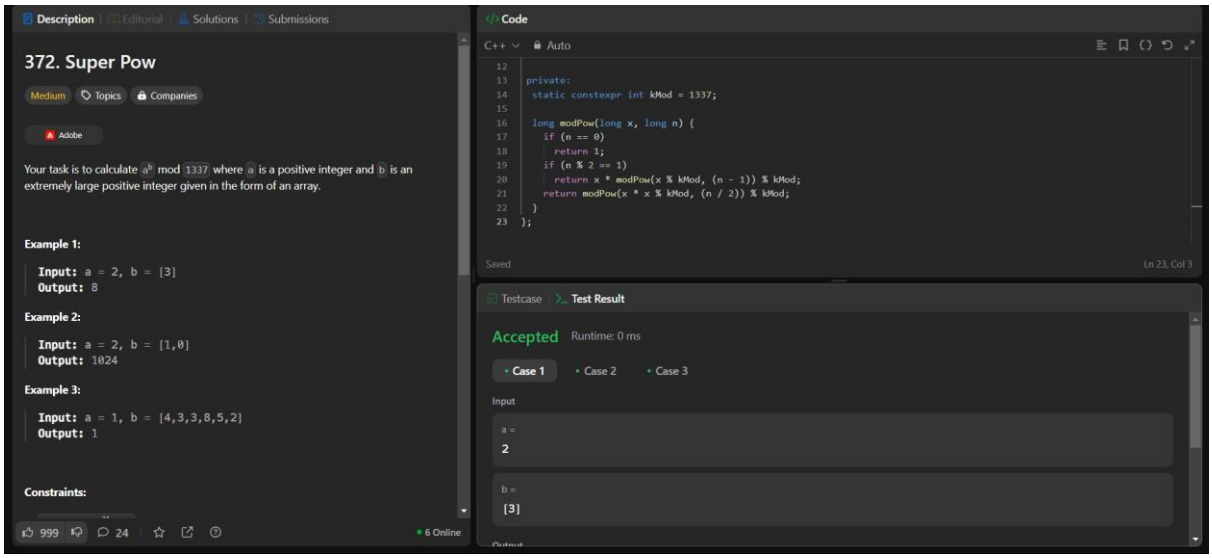
4. Maximum Subarray:



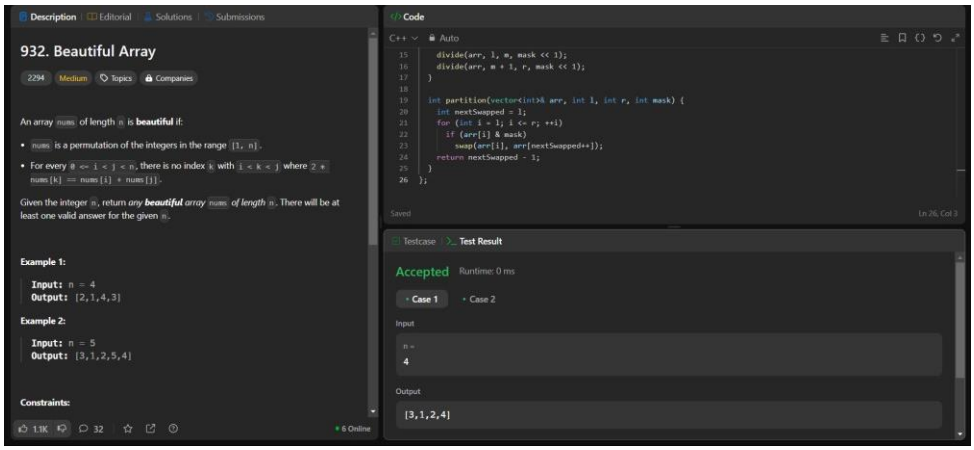
5. Search a 2D Matrix II:



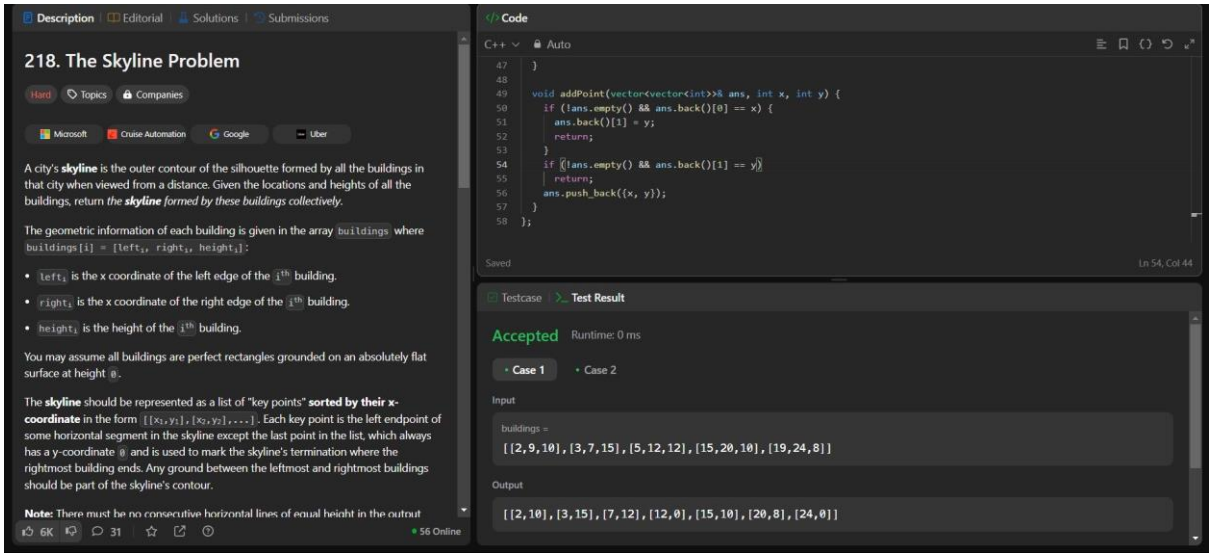
6. Super Pow:



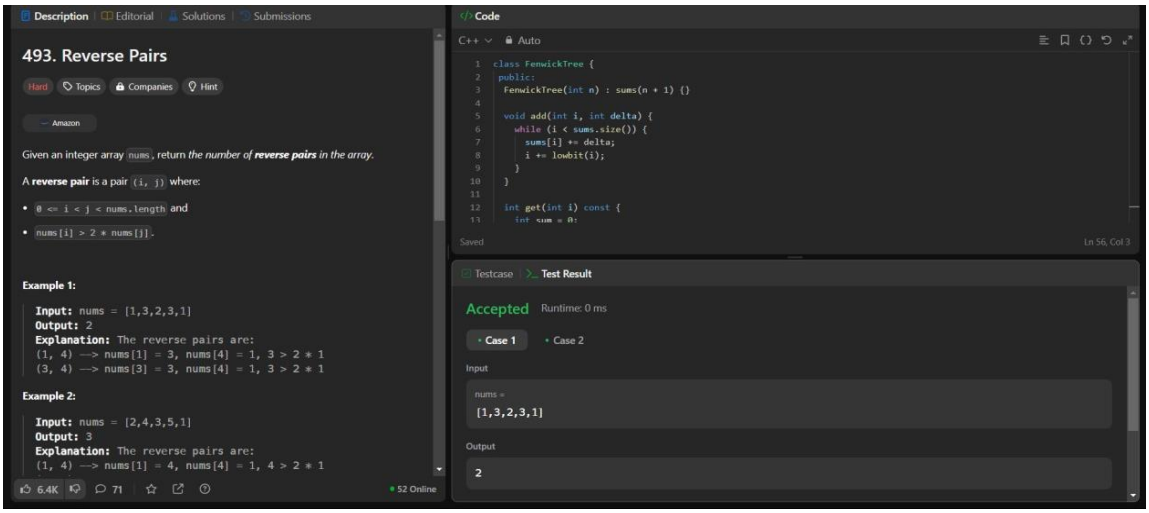
7. Beautiful Array:



8. The Skyline Problem:



9. Reverse Pairs:



10.Longest Increasing Subsequence II: