Advanced Pragramming

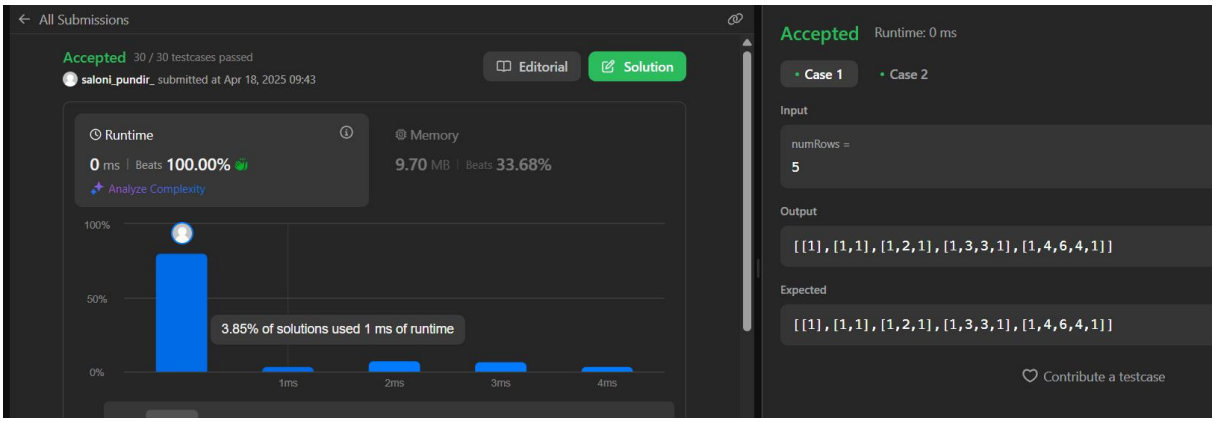
ASSIGNMENT 10

Q1. Pascal's Triangle.

Code:

```cpp
class Solution {
public:
    vector<vector<int>> generate(int numRows) {
        vector<vector<int>> output(numRows);
        for(int i=0; i<numRows; i++){
            output[i].resize(i+1, 1);
            for(int j=1; j<i; j++){
                output[i][j] = output[i-1][j-1] + output[i-1][j];
            }
        }
        return output;
    }
};
```

Output:



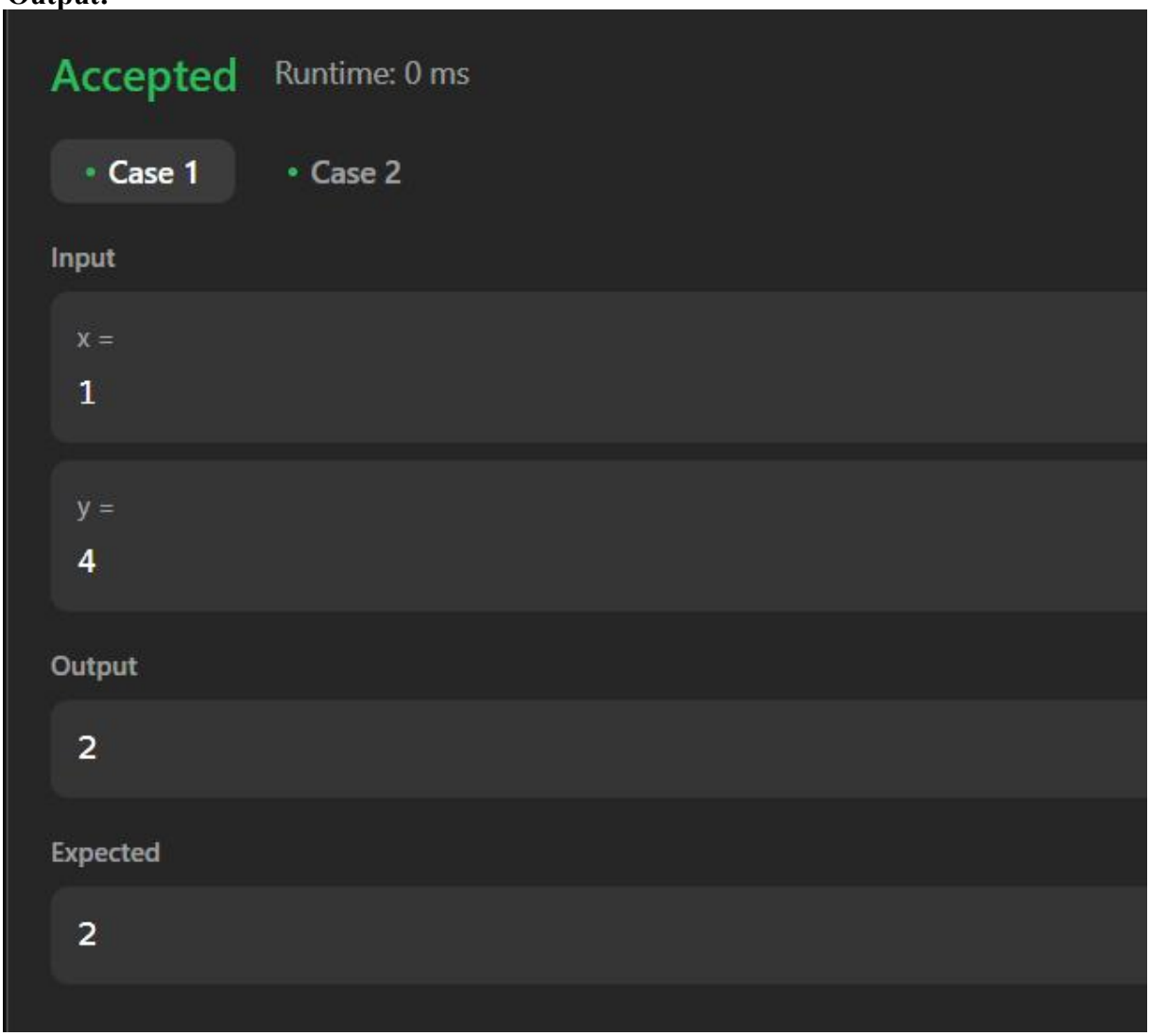
Q2.hamming distance.

Code:

C++ Auto

```
class Solution {
public:
    int hammingDistance(int x, int y) {
        return __builtin_popcount(x^y);
    }
};
```

Output:

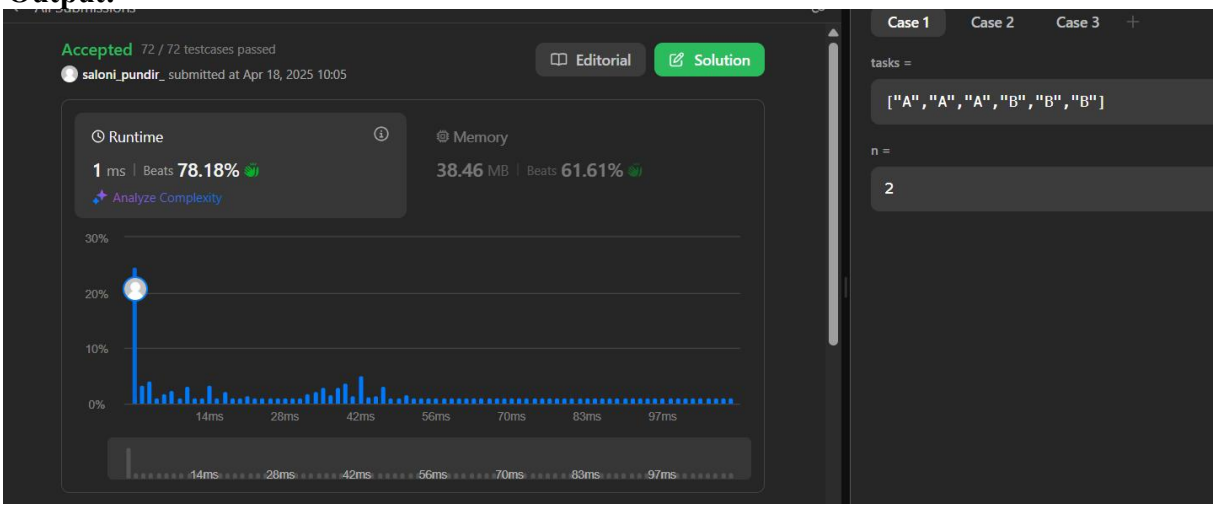


Q3. Task Scheduler.

Code:

```cpp
class Solution {
public:

int rem(vector<int> map, int T){
    int sum = 0;
    for(int i: map ){
        if(i>T){
            sum += i - T;
        }
    }
    return sum;
}

    int leastInterval(vector<char>& task, int n) {
        if(task.size() == 1) return 1;

        vector<int> map(26,0);
        int t = 0;
        unordered_set<char> s;
        for(char ch: task){
            map[ch-'A']++;
            s.insert(ch);
        }

        int greatest = -1, ind = -1;
        for(int i= 0; i<26; i++){
            if(map[i]> greatest){
                greatest = map[i];
                ind = i:
```

Output:



Q4. Number of 1 Bits.

Code:

C++ Auto

```cpp
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        for (int i = 0; i < 32; i++) {
            if ((n >> i) & 1) {
                res += 1;
            }
        }
        return res;
    }
};
```

Output: