Course Name:	**Competitive Programming Laboratory (216U01L401)**	**Semester:**	**IV**
Date of Performance:	**03 / 04 / 2025**	**DIV/ Batch No:**	**A1**
Student Name:	**Aaryan Sharma**	**Roll No:**	**16010123012**

Experiment No: 7

Title: **To use a Game theory approach to solve competitive programming problems.**

Aim and Objective of the Experiment:

1. **Understand** the concepts of geometry-based approach.
2. **Apply** the concepts to solve the problem.
3. **Implement** the solution to given problem statement.
4. **Create** test cases for testing the solution.
5. **Analyze** the result for efficiency of the solution.

COs to be achieved:

CO4: Apply geometry, game theory, and string algorithms in competitive programming.

Books/ Journals/ Websites referred:

1. https://cp-algorithms.com/game_theory/sprague-grundy-nim.html

Theory:

Game theory is a mathematical framework used for analyzing competitive situations where the outcome for each participant depends on the choices of others. One classic game theory problem is the Nim Game, which involves players taking turns to remove stones from a heap.
The game follows these rules:
1. There is a single heap with n stones.
2. Two players take turns to remove 1, 2, or 3 stones per move.
3. The player who removes the last stone wins.

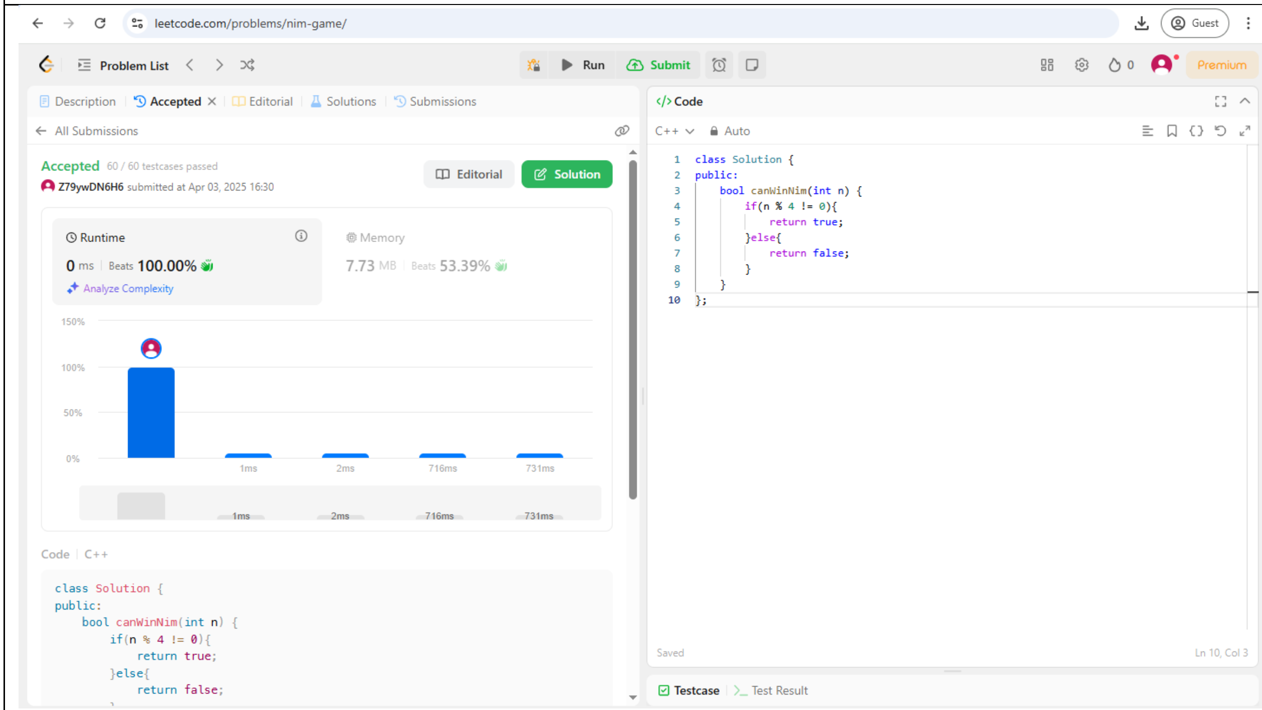
The key concept used to determine the winner is the Sprague-Grundy Theorem, which states:
1. If n % 4 == 0, the player to move will lose if both play optimally.
2. Otherwise, the first player has a winning strategy.

Problem statement
You are playing the following Nim Game with your friend: Initially, there is a heap of stones on the table. You and your friend will alternate taking turns, and you go first. On each turn, the person whose turn it is will remove 1 to 3 stones from the heap. The one who removes the last stone is the winner. Given n, the number of stones in the heap, return true if you can win the game assuming both you and your friend play optimally, otherwise return false. Example 1: Input: n = 4 Output: false Explanation: These are the possible outcomes: 1. You remove 1 stone. Your friend removes 3 stones, including the last stone. Your friend wins. 2. You remove 2 stones. Your friend removes 2 stones, including the last stone. Your friend wins. 3. You remove 3 stones. Your friend removes the last stone. Your friend wins. In all outcomes, your friend wins. Example 2: Input: n = 1 Output: true Example 3: Input: n = 2 Output: true

Code :

```
class Solution {
public:
    bool canWinNim(int n) {
        if(n % 4 != 0){
            return true;
        }else{
            return false;
        }
    }
};
```

Output:



Conclusion:

I have applied game theory principles to solve this problem efficiently. I analyzed the Nim Game and implemented an optimal solution using a simple mathematical observation (n % 4). The experiment demonstrated how theoretical concepts from game theory can be used to design winning strategies in competitive programming.