R.SARANYA
2300033443

DAA SKILL-12

Lego Blocks

```java
import java.util.Scanner;

public class legoBlocks {
  private int M = 0, N = 0;
  long[] table = new long[1];
  long[] table2 = new long[1];

  public legoBlocks(int m, int n) {
    M = m;
    N = n;
    table = new long[N + 1];
    table2 = new long[N + 1];
    for (int i = 0; i <= N; i++) table2[i] = -1;
    filltable2(N);

    for (int i = 0; i <= N; i++) {
      long res = 1;
      for (int j = 0; j < M; j++) {
        res = (res * table2[i]) % 1000000007;
      }
      table2[i] = res;
    }
  }
```

```java
    for (int i = 0; i <= N; i++) {
        table[i] = -1;
    }
}

private long filltable2(int n) {
    if (n < 0) {
        return 0; // Added return statement for negative values of `n`
    }

    if (table2[n] == -1) {
        if (n == 0) {
            table2[n] = 1;
        } else {
            table2[n] = (filltable2(n - 1) + filltable2(n - 2) + filltable2(n - 3) +
filltable2(n - 4)) % 1000000007;
        }
    }
    return table2[n];
}

private long evaluate() {
    long result = helper(N);
    return result; // Added return statement
}
```

```java
private long helper(int n) {
    if (table[n] == -1) {
        if (n == 1) {
            table[n] = 1;
        } else {
            table[n] = table2[n];
            for (int i = 1; i < n; i++) {
                table[n] = (table[n] - helper(n - i) * table2[i]) % 1000000007;
            }
            if (table[n] < 0) {
                table[n] += 1000000007;
            }
        }
    }
    return table[n];
}

public static void main(String[] args) {
    Scanner inp = new Scanner(System.in);
    int testcases = inp.nextInt();

    for (int i = 0; i < testcases; i++) {
        int m = inp.nextInt();
        int n = inp.nextInt();
        legoBlocks T = new legoBlocks(m, n);
        System.out.println(T.evaluate());
```
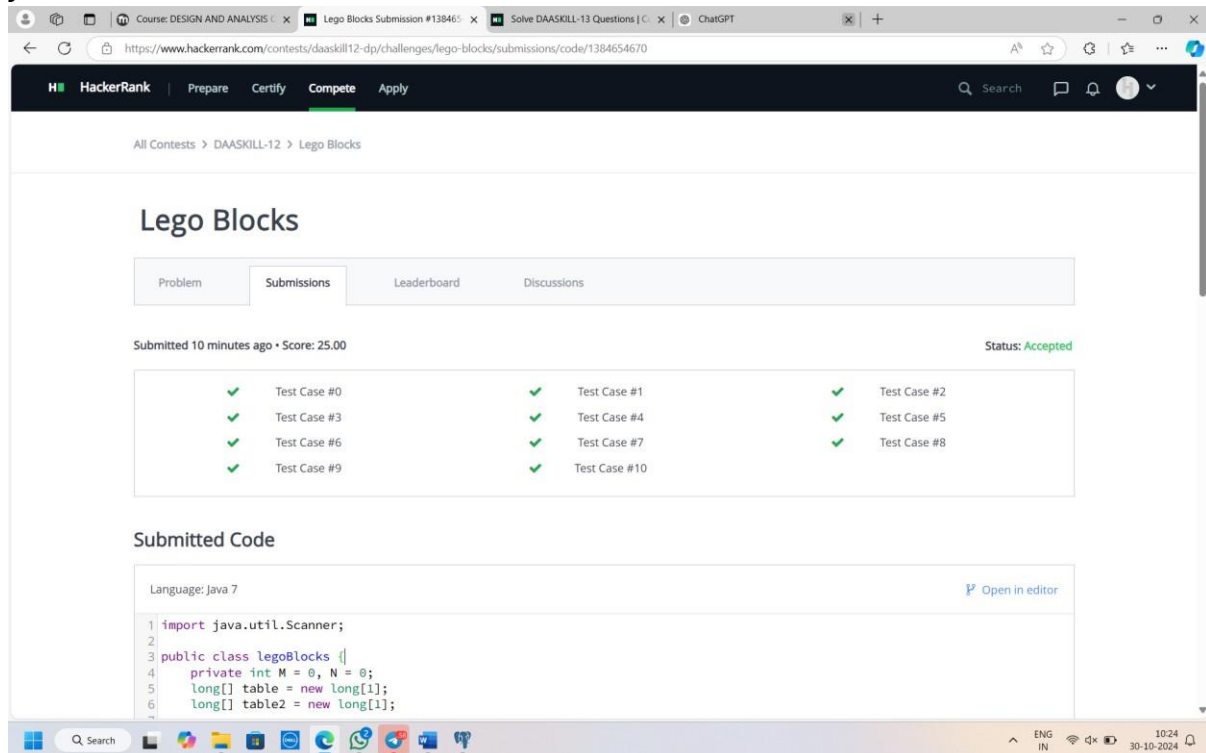
```
        }

        inp.close();

    }

}
```


Stock Maximize

```java
import java.util.Scanner;

public class Solution {

    private Scanner in;

    public Solution() {
        in = new Scanner(System.in);
    }
```

```java
public void run() {
    int testCases = in.nextInt();

    for (int test = 0; test < testCases; test++) {
        int n = in.nextInt();
        int[] arr = new int[n];
        int[] dp = new int[n];

        // Read array values
        for (int i = 0; i < n; i++) {
            arr[i] = in.nextInt();
        }

        // Initialize max to the last element in arr
        int max = arr[n - 1];

        // Fill dp array from right to left with the maximum values
        for (int i = n - 1; i >= 0; i--) {
            max = Math.max(arr[i], max);
            dp[i] = max;
        }

        // Calculate the sum of differences between dp[i] and arr[i]
        long sum = 0;
        for (int i = 0; i < n; i++) {
            sum += (dp[i] - arr[i]);
```

```java
        }

        // Print the result for the current test case
        System.out.println(sum);
    }
}


public static void main(String[] args) {
    Solution solution = new Solution();
    solution.run();
}
}
```