

ID NO : 2300031672

NAME : K.Imran

LAB SKILL-WEEK 12

1) Lego Blocks:

All Contests > DAASKILL-12 > Lego Blocks

Lego Blocks

Problem

Submissions

Leaderboard

Discussions

Submitted a few seconds ago • Score: 25.00

Status: Accepted

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10		

Submitted Code:-

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define MOD 1000000007
```

```
void calculateRowCombinations(int maxWidth, int *rowCombinations) {  
    rowCombinations[0] = 1;
```

```
    for (int i = 1; i <= maxWidth; i++) {  
        rowCombinations[i] = rowCombinations[i - 1];  
        if (i >= 2) rowCombinations[i] = (rowCombinations[i] + rowCombinations[i - 2]) % MOD;  
        if (i >= 3) rowCombinations[i] = (rowCombinations[i] + rowCombinations[i - 3]) % MOD;  
        if (i >= 4) rowCombinations[i] = (rowCombinations[i] + rowCombinations[i - 4]) % MOD;  
    }  
}
```

```
}
```

```
int legoBlocks(int n, int m) {  
    int *rowCombinations = (int *)malloc((m + 1) * sizeof(int));  
    int *totalWays = (int *)malloc((m + 1) * sizeof(int));  
    int *solidWays = (int *)malloc((m + 1) * sizeof(int));
```

```
    calculateRowCombinations(m, rowCombinations);
```

```
    for (int i = 1; i <= m; i++) {  
        totalWays[i] = 1;  
        for (int j = 0; j < n; j++) {  
            totalWays[i] = (1LL * totalWays[i] * rowCombinations[i]) % MOD;
```

```

    }
}

for (int i = 1; i <= m; i++) {
    solidWays[i] = totalWays[i];
    for (int j = 1; j < i; j++) {
        solidWays[i] = (solidWays[i] - (1LL * solidWays[j] * totalWays[i - j]) % MOD + MOD) %
MOD;
    }
}

int result = solidWays[m];
free(rowCombinations);
free(totalWays);
free(solidWays);

return result;
}

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int n, m;
        scanf("%d %d", &n, &m);
        printf("%d\n", legoBlocks(n, m));
    }
    return 0;
}

```

2) Stock Maximize:

[Prepare](#)
[Certify](#)
[Compete](#)
[Apply](#)

Search

[All Contests](#) > [DAASKILL-12](#) > [Stock Maximize](#)

Stock Maximize

Problem

Submissions

Leaderboard

Discussions

 Submitted a few seconds ago • Score: 25.00
 Status: Accepted

Submitted Code:

```
#include <stdio.h>
#include <stdlib.h>

long stockmax(int prices_count, int* prices) {
    long profit = 0;
    int max_future_price = 0;

    for (int i = prices_count - 1; i >= 0; i--) {
        if (prices[i] > max_future_price) {
            max_future_price = prices[i];
        }

        profit += max_future_price - prices[i];
    }

    return profit;
}

int main() {
    int t;
    scanf("%d", &t);

    for (int t_itr = 0; t_itr < t; t_itr++) {
        int n;
        scanf("%d", &n);

        int* prices = (int*)malloc(n * sizeof(int));
        for (int i = 0; i < n; i++) {
            scanf("%d", &prices[i]);
        }

        long result = stockmax(n, prices);
        printf("%ld\n", result);

        free(prices);
    }

    return 0;
}
```