Connecting Towns

locked

Problem | Submissions | Leaderboard | Discussions

Gandalf is travelling from **Rohan** to **Rivendell** to meet Frodo but there is no direct route from **Rohan** (T_1) to **Rivendell** (T_n).

But there are towns T_2,T_3,T_4...T_{n-1} such that there are N_1 routes from Town T_1 to T_2, and in general, N_i routes from T_i to T_{i+1} for i=1 to n-1 and 0 routes for any other T_i to T_j for $j \neq i+1$

Find the total number of routes Gandalf can take to reach Rivendell from Rohan.

Note

Gandalf has to pass all the towns T_i for i=1 to n-1 in numerical order to reach T_n.
For each T_i , T_{i+1} there are only N_i distinct routes Gandalf can take.

Input Format

The first line contains an integer T, T test-cases follow.
Each test-case has 2 lines. The first line contains an integer N (the number of towns).
The second line contains N - 1 space separated integers where the i^{th} integer denotes the number of routes, N_i, from the town T_i to T_{i+1}

Output Format

Total number of routes from T_1 to T_n modulo 1234567

http://en.wikipedia.org/wiki/Modular_arithmetic

Constraints

$1 <= T <= 1000$

$2 < N <= 100$

$1 <= N_i <= 1000$

Sample Input

```
2
3
1 3
4
2 2 2
```

Sample Output

```
3
8
```

Explanation

Case 1: 1 route from T_1 to T_2, 3 routes from T_2 to T_3, hence only 3 routes.

Case 2: There are 2 routes from each city to the next, at each city, Gandalf has 2 choices to make, hence 2 * 2 * 2 = 8.

Submissions: 0

Max Score: 10

Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable)

Python 3

```python
#!/bin/python3

import os
import sys

#
# Complete the connectingTowns function below.
#
def connectingTowns(n, routes):
    while n > 0:
        res = 0
        c = 0
        while c <len(routes):
            if res == 0:
                res = routes[c]
            else:
                res = res * routes[c]
                res = res % 1234567
            c = c + 1


        n = n - 1
    return res

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    t = int(input())

    for t_itr in range(t):
        n = int(input())
```

```
32
33          routes = list(map(int, input().rstrip().split()))
34
35          result = connectingTowns(n, routes)
36
37          fptr.write(str(result) + '\n')
38
39      fptr.close()
40
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
2
3
1 3
4
2 2 2
```

Your Output (stdout)

```
3
8
```

Expected Output

```
3
8
```