

Construct the Array ☆

302.41 more points to get your next star!

Rank: 24088 | Points: 1897.59/2200



Problem

Submissions

Leaderboard

Discussions

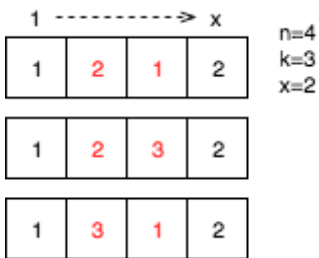
Editorial

Your goal is to find the number of ways to construct an array such that consecutive positions contain different values.

Specifically, we want to construct an array with n elements such that each element between 1 and k , inclusive. We also want the first and last elements of the array to be 1 and x .

Given n , k and x , find the number of ways to construct such an array. Since the answer may be large, only find it modulo $10^9 + 7$.

For example, for $n = 4$, $k = 3$, $x = 2$, there are 3 ways, as shown here:



Complete the function countArray which takes input n , k and x . Return the number of ways to construct the array such that consecutive elements are distinct.

Constraints

- $3 \leq n \leq 10^5$
- $2 \leq k \leq 10^5$
- $1 \leq x \leq k$

Subtasks

- For 20% of the maximum score, $n \leq 10^3$ and $k \leq 10^2$

Sample Input

$n = 4, k = 3, x = 2$

Sample Output

3

Explanation

Refer to the diagram in the challenge statement.

Author kemalkochekov

Difficulty Medium

Max Score 35

Submitted By 4158

NEED HELP?

 [View discussions](#)

 [View editorial](#)

 [View top submissions](#)

RATE THIS CHALLENGE



MORE DETAILS

 [Download problem statement](#)

 [Download sample test cases](#)

 [Suggest Edits](#)



C++14



```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<string> split_string(string);
6
7 // Complete the countArray function below.
```

```

8  long countArray(int n, int k, int x) {
9      // Return the number of ways to fill in the array.
10
11  }
12
13  int main()
14  {
15      ofstream fout(getenv("OUTPUT_PATH"));
16
17      string nkx_temp;
18      getline(cin, nkx_temp);
19
20      vector<string> nkx = split_string(nkx_temp);
21
22      int n = stoi(nkx[0]);
23
24      int k = stoi(nkx[1]);
25
26      int x = stoi(nkx[2]);
27
28      long answer = countArray(n, k, x);
29
30      fout << answer << "\n";
31
32      fout.close();
33
34      return 0;
35  }
36
37  vector<string> split_string(string input_string) {
38      string::iterator new_end = unique(input_string.begin(), input_string.end(), []
39      (const char &x, const char &y) {
          return x == y and x == ' ';

```

Line: 1 Col: 1

 [Upload Code as File](#)

☐ Test against custom input

Run Code

Submit Code