



## Experiment-8

**Student Name: Jitesh Kumar**

**Branch: BE-CSE**

**Semester: 5**

**Subject Name: Competitive Coding**

**UID: 20BCS2334**

**Section/Group: WM\_903-A**

**Subject Code: 20CSP-317**

### Problem 8.1(Construct the Array)

**Aim:** 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

elements such that each element between and , inclusive. We also want the first and last elements of the array to be 1 and x .

You are given a function,

### CODE:

```
const int MOD = 1000000007;
long countArray(int n, int k, int x) {
    // Return the number of pro to fill in the array.
    long pro[2][2];
    pro[0][0] = 1;
    pro[0][1] = 0;
    bool sec = true;
    for (int i = 0; i < n-1; i++) {
        pro[sec][0] = (pro[!sec][1] * (k - 1)) % MOD;
        pro[sec][1] = (pro[!sec][1] * (k -
2) + pro[!sec][0]) % MOD;
        sec = !sec;
    }
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
long ans;  
if (x == 1) {  
    ans = (pro[sec][1] * (k - 1)) % MOD;  
}  
else {  
    ans = (pro[sec][1] * (k - 2) + pro[sec][0]) % MOD;  
}  
return ans;  
}
```

## OUTPUT:

### Congratulations

You solved this challenge. Would you like to challenge your friends?

[f](#) [t](#) [in](#)

[Next Challenge](#)

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

✓ Test case 6

Compiler Message

Success

Input (stdin)[Download](#)

1	4 3 2
---	-------

Expected Output[Download](#)

1	3
---	---

## Problem Sam and Substring

**Aim:** Samantha and Sam are playing a numbers game. Given a number as a string, no leading zeros, determine the sum of all integer values of substrings of the string.

Given an integer as a string, sum all of its substrings cast as integers. As the number may become large, return the value modulo

CODE:

```
long long mod = 1000000007;

int substrings(string s) {
    long long arr1[s.length() + 1];
    arr1[0] = 1;
    for (int i = 1; i <= s.length(); ++i) {
        arr1[i] = arr1[i - 1] * 10 % mod;
    }
    long long sum[s.length() + 1];
    sum[0] = arr1[0];
    for (int i = 1; i <= s.length(); ++i) {
        sum[i] = sum[i - 1] + arr1[i];
        sum[i] %= mod;
    }
    long long ans = 0;
    int n = s.length();
    for (int i = 0; i < n; ++i) {
        ans += (s[i] - '0') * (i + 1) * sum[n - 1 - i];
        ans %= mod;
    }
    return ans;
}
```

## OUTPUT:

### Congratulations

You solved this challenge. Would you like to challenge your friends?

[f](#) [t](#) [in](#)

[Next Challenge](#)

✓ Test case 0

✓ Test case 1

✓ Test case 2

✓ Test case 3

✓ Test case 4

✓ Test case 5

✓ Test case 6

Compiler Message

Success

Input (stdin) [Download](#)

1	16
---	----

Expected Output [Download](#)

1	23
---	----

## Learning Outcomes:

1. Learn About The Dynamic Programming
2. Learn About Recursion
3. Learn How to Reduce the time complexity using DP