2 Questions Total Marks: 200.0

2 Programming Questions

1. Min-Max Weighted Edge + 100.0

2. Colorful Buildings + 100.0

Question 2 Max. Marks 100.00

Colorful Buildings

There are N buildings in a row. Each of these buildings needs to be painted by one of the K colors. Buildings look beautiful only if no adjacent buildings are painted with the same color. Find the number of ways to paint these buildings such that they are beautiful. Since the number could be very large output it modulo $10^9 + 7$.

Input Format

The first line of the input contains an integer T, the total number of test cases.
Then T lines follow, each containing two space-separated integers N and K, the total number of buildings and the number of colors available.

Output Format

For each test case output the number of ways to paint the buildings such that they are beautiful modulo $10^9 + 7$.

Constraints

$1 \leq T \leq 10$
$1 \leq N \leq 10^5$
$2 \leq K \leq 10^5$

Sample Input	Sample Output
2 4 2 4 6	2 750

Explanation

Sample test case 1 : Let A and B be the two available colors then there are just two possible ways of coloring the buildings as ABAB and BABA.

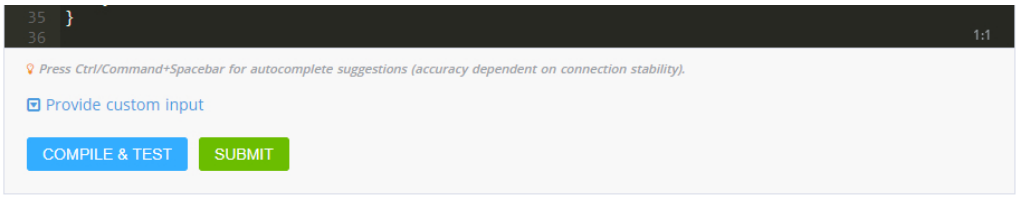
Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Marks are awarded if any testcase passes
Allowed Languages: Bash, C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), TypeScript, Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic

New Submission | All Submissions

Java 8 (oracle 1.8.0_131) Save

```java
/* IMPORTANT: Multiple classes and nested static classes are supported */

/*
 * uncomment this if you want to read input.
//imports for BufferedReader
import java.io.BufferedReader;
import java.io.InputStreamReader;

//import for Scanner and other utility classes
import java.util.*;
*/

// Warning: Printing unwanted or ill-formatted data to output will cause the test cases to fail

class TestClass {
    public static void main(String args[] ) throws Exception {
        /* Sample code to perform I/O:
         * Use either of these methods for input

        //BufferedReader
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String name = br.readLine();                // Reading input from STDIN
        System.out.println("Hi, " + name + ".");    // Writing output to STDOUT

        //Scanner
        Scanner s = new Scanner(System.in);
        String name = s.nextLine();                 // Reading input from STDIN
        System.out.println("Hi, " + name + ".");    // Writing output to STDOUT

        */

        // Write your code here

    }
```


35 }
36
1:1
Press Ctrl/Command+Spacebar for autocomplete suggestions (accuracy dependent on connection stability).
Provide custom input
COMPILE & TEST
SUBMIT