|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| http://ace.delos.com/usaco/cowhead2.gif | |  | | --- | | Contest: FEB11 **GOLD** Division | |  | |  | |  | |

**ANALYSIS MODE  
Submit solutions for your own enjoyment.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**GOLD PROBLEMS**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Three problems numbered 1 through 3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 1: Cowlphabet [Michael Cohen, 2010]**

**Like all bovines, Farmer John's cows speak the peculiar 'Cow'**

**language. Like so many languages, each word in this language comprises**

**a sequence of upper and lowercase letters (A-Z and a-z). A word**

**is valid if and only if each ordered pair of adjacent letters in**

**the word is a valid pair.**

**Farmer John, ever worried that his cows are plotting against him,**

**recently tried to eavesdrop on their conversation. He overheard one**

**word before the cows noticed his presence. The Cow language is**

**spoken so quickly, and its sounds are so strange, that all that**

**Farmer John was able to perceive was the total number of uppercase**

**letters, U (1 <= U <= 250) and the total number of lowercase**

**letters, L (1 <= L <= 250) in the word.**

**Farmer John knows all P (1 <= P <= 200) valid ordered pairs of**

**adjacent letters. He wishes to know how many different valid**

**words are consistent with his limited data. However, since**

**this number may be very large, he only needs the value modulo**

**97654321.**

**PROBLEM NAME: cowlpha**

**INPUT FORMAT:**

**\* Line 1: Three space-separated integers: U, L and P**

**\* Lines 2..P+1: Two letters (each of which may be uppercase or**

**lowercase), representing one valid ordered pair of adjacent**

**letters in Cow.**

**SAMPLE INPUT (file cowlpha.in):**

**2 2 7**

**AB**

**ab**

**BA**

**ba**

**Aa**

**Bb**

**bB**

**INPUT DETAILS:**

**The word Farmer John overheard had 2 uppercase and 2 lowercase**

**letters. The valid pairs of adjacent letters are AB, ab, BA, ba,**

**Aa, Bb and bB.**

**OUTPUT FORMAT:**

**\* Line 1: A single integer, the number of valid words consistent with**

**Farmer John's data mod 97654321.**

**SAMPLE OUTPUT (file cowlpha.out):**

**7**

**OUTPUT DETAILS:**

**The possible words are:**

**AabB**

**ABba**

**abBA**

**BAab**

**BbBb**

**bBAa**

**bBbB**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 2: The Lost Cows [Adapted, 2010]**

**One sunny day farmer John was kidnapped by evil farmer Marcus's**

**cows. FJ wasn't too concerned about his forced holiday but wanted**

**to make sure that his cows got home safely together.**

**The cows are spread out in every one of FJ's N (3 <= N <= 200)**

**pastures conveniently numbered 1..N. The barn is located at pasture**

**1. The farm has an interesting navigation system: at every pasture**

**i there are M (1 <= M <= 200) signs S\_ij (1 <= S\_ij <= N) which one**

**could reference as S\_i1..S\_iM; each sign points the way to a pasture.**

**Sometimes a sign points to a path that leads back to the same**

**pasture.**

**Farmer Marcus's cows allow FJ to write a single message to all of**

**his cows. FJ's plan is to write a list of sign numbers such that**

**any cow who follows those instructions will all arrive at the barn**

**when each cow has completed all the instructions.**

**When a cow starts at a given pasture then she will first follow the**

**path indicated by the first sign number on FJ's list. When she**

**arrives at the second pasture, she looks at the second sign of FJ's**

**list and follows the path marked by that sign. She continues until**

**she exhausts the instruction list, at which point she should be at**

**the barn.**

**Find a list of instructions containing no more than 5,000,000 sign**

**numbers that will guide every cow, from every pasture, to the barn**

**after all instructions are followed. It is guaranteed that such a**

**list exists.**

**Consider a set of three signs in four pastures that direct the cows**

**like these do:**

**\*\* Pasture# \*\***

**1 2 3 4**

**Sign 1 4 4 1 3**

**Sign 2 1 3 2 4**

**Sign 3 4 2 3 1**

**The set of instructions below will direct cows to the barn from any**

**of the four pastures:**

**Instruction# Sign# Instruction# Sign#**

**1 1 5 3**

**2 2 6 1**

**3 1 7 3**

**4 2**

**The cow in pasture 1 will read sign #1 at time 1 and be directed**

**to pasture 4. At time 2, she is in pasture 4 and (per FJ's**

**instructions) read sign #2 and then be directed to pasture 4. Below**

**is a table that shows the cow's travels:**

**\* \* \* \* Cow in pasture 1 \* \* \* \***

**Time CurrentPasture# WhichSign Sign->Nextpasture**

**1 1 1 4**

**2 4 2 4 (same pasture!)**

**3 4 1 3**

**4 3 2 2**

**5 2 3 2 (same pasture)**

**6 2 1 4**

**7 4 3 1 Barn!**

**Similarly: Pasture 2's cow visits pastures [2]-4-4-3-2-2-4-1.**

**Pasture 3's cow visits pastures [3]-1-1-4-4-1-4-1.**

**Pasture 4's cow visits pastures [4]-3-2-4-4-1-4-1.**

**Given a set of signs, create a set of instructions.**

**PROBLEM NAME: lostcows**

**INPUT FORMAT:**

**\* Line 1: Two space separated integers: N and M**

**\* Lines 2..M+1: Line i+1 describes the contents of each pasture's N**

**signs with N integers: S\_1i..S\_Ni**

**SAMPLE INPUT (file lostcows.in):**

**4 3**

**4 4 1 3**

**1 3 2 4**

**4 2 3 1**

**OUTPUT FORMAT:**

**\* Lines 1..?: The sign numbers the cows should follow, one per line.**

**SAMPLE OUTPUT (file lostcows.out):**

**1**

**2**

**1**

**2**

**3**

**1**

**3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 3: Generic Cow Protests [Neal Wu, 2010]**

**Farmer John's N (1 <= N <= 100,000) cows are lined up in a row and**

**numbered 1..N. The cows are conducting another one of their strange**

**protests, so each cow i is holding up a sign with an integer A\_i**

**(-10,000 <= A\_i <= 10,000).**

**FJ knows the mob of cows will behave if they are properly grouped**

**and thus would like to arrange the cows into one or more contiguous**

**groups so that every cow is in exactly one group and that every**

**group has a nonnegative sum.**

**Help him count the number of ways he can do this, modulo 1,000,000,009.**

**By way of example, if N = 4 and the cows' signs are 2, 3, -3, and**

**1, then the following are the only four valid ways of arranging the**

**cows:**

**(2 3 -3 1)**

**(2 3 -3) (1)**

**(2) (3 -3 1)**

**(2) (3 -3) (1)**

**Note that this example demonstrates the rule for counting different**

**orders of the arrangements.**

**PROBLEM NAME: protest**

**INPUT FORMAT:**

**\* Line 1: A single integer: N**

**\* Lines 2..N + 1: Line i + 1 contains a single integer: A\_i**

**SAMPLE INPUT (file protest.in):**

**4**

**2**

**3**

**-3**

**1**

**OUTPUT FORMAT:**

**\* Line 1: A single integer, the number of arrangements modulo**

**1,000,000,009.**

**SAMPLE OUTPUT (file protest.out):**

**4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***