Problem A: Palentine's Day

Setter: Alina Zaman
Alternate Writer: Md. Shiplu Hawlader
Type: Giveaway

You have to print 365*366 / 2.

Problem B: Special Days

Setter: Md. Shiplu Hawlader
Alternate Writer: Hasnain Heickal
Type: Giveaway, Ad Hoc

Little bit of tedious to implement. Just loop over the sorted special day array and stop if you surpass a day.

Problem C: Bulb and Pipes

Setter: Shafaet Ashraf
Alternate Writer: Aninda Majumdar
Type: Greedy, observation, adhoc

Draw small dots in the pipe. Start from the beginning of each pipe and draw the dots at each r distance. (if the next endpoint is less than r distance away, draw the dot at the endpoint). Now think of each dot as a node which is connected to the next dot. Now place bulbs in alternate nodes starting from the 2nd node. Answer is number of nodes /2.

There is a tricky case. If there are multiples pipes with same direction, you must treat them as one single pipe. This is the tricky input:

```
2
4 2
5 5 5 5
R R R R
1 2
20
R
```

Answer for both case should be 5. If you are getting 6 in the first case, you are missing the trick.

Problem D: Sum of Two Sequences

Setter: Mohammad Samiul Islam
Alternate Writer: Mehdi Rahman
Type: Simple Math

Just use arithmetic progression formula to calculate the sums.

Problem E: Coin and Bulbs

Setter: Shafaet Ashraf
Alternate Writer: Nikoloz Svanidze
Type: Expected value, probability

Suppose there n bulbs and x bulbs are turned on at this moment. Lets define r = n-x (remaining number of bulbs), s=1-p (probability of getting a head.

Now chance of choosing a already turned on bulb is x/n. In this case, we will need (x/n)(1+f(x)) number of moves. Chance of choosing a off bulb is r/n. In case we will need (r/n)(p(1+f(x))+s(1+f(x+1)) number of moves. So we can say f(x)=(x/n)(1+f(x)) + (r/n)(p(1+f(x))+s(1+f(x+1)). Simplify the equation by moving f(x) to the left and you will get a recursive equation. You need to be skilled with dealing fractions too to solve the problem.

Problem F: Grundy and K-Nim

Setter: Hasnain Heickal
Alternate Writer: Md. Shiplu Hawlader
Type: Game Theory

To solve this problem we need to find grundy number for each pile.
The grundy number of pile with X stone is actually X%(k+1).
After that we just need to XOR grundy numbers from all piles.

Proof:
A pile with 0 to k stones will have grundy number equal to that.
A pile with k+i stones (where 0<i<=k) we will reach all the values from [0,k] except i-1 so its mex will be i-1.
Which is (k+i)%(k+1), since (k+i)=(k+1 + i-1)

Problem G: Merge the Strings

Setter: Aninda Majumdar
Alternate Writer: Mohammad Samiul Islam
Type: DP

Typical LCS type DP which will return a **pair<int, string>** because both the length and string need to be calculated separately. We need to return the string to compare lexicographically.

Problem H: Count Clog

Setter: Mehdi Rahman
Alternate Writer: Mohammad Samiul Islam
Type: Combinatorics, Number Theory

The intended solution is using Inclusion-Exclusion principle. The formulation is similar to counting derangement. In addition, the modulo is not prime and the prime factors of modulo may not be co-prime with the denominator when you need its modular inverse.
This can be solved by separating the prime factors of modulo from the numerator and denominator. Suppose, one needs to calculate (315/45) modulo 5. Here 45 and 5 are not coprime, but assuredly numerator is divisible by denominator. It can be reformed as,

(63*5)/(9*5) modulo 5
= 63/9 modulo 5, since the prime components are separated, the modular inverse of denominator exists now.

Problem I: Pom Gana

Setter: Md. Shiplu Hawlader
Alternate Writer: Hasnain Heickal
Type: Graph Theory

We have to find how many nodes in the graph are not part of any cycle. There are several ways using bfs, dfs etc but the most easiest one with this limit is Floyd-Warshall. Start with a boolean matrix with 1 in the cell(x, y) if there is an edge from x to y. Set all the diagonal to 0. Now run Floyd-Warshall with relaxation g[i][j] = g[i][j] || g[i][k] && g[k][j]. After finishing Warshall, if g[i][i] contains 0 then it is not included in any cycle because there is no path from i to i.