Hello world,

I hope you guys enjoyed my problem set :)
Yo yes, I am the problem setter this time!
Rate it **HERE**

** I hour into the contest, I started to write this document. This is how the discussion between me and anup is.
Me (Before contest started): I am expecting 20 teams to solve all 5.

Me(1 hour into contest): Good so far.. looks like i again under estimated difficulty :(
Anup: you had expected 20 teams to solve 5 :P
Me: Yeah Anup, I go back on that now, I will be happy to see 5 teams solve all 5 :)
Anup: and i will be happy to see 1 team solve all 5 :)
Me: I am underestimating difficulty, you are underestimating teams :P
Anup: Hehe, no i am just inferring from the stats :P
Mahbub: Frankly, I dont expect any team to solve all (or may be 1-2 but I doubt)

That was it. Let us see how it ends up :)

Here is a short writeup of problem solutions.

I would like to thank **Mr.X** (Remember? The Anon guy from Amritha regionals team), He helped us when testing of set was going on.

The last cracker
Author: Anudeep Nekkanti
Tester: Mahbubul Hasan
Cakewalk, You need to print the value after M rounds. Answer = ((N-1+M)%N) + 1.
Constraints are made so that simply looping M times works.
Link to solution: http://ideone.com/r8jEnK

Discount on crackers
Author: Anudeep Nekkanti
Tester: Mahbubul Hasan
In this problem, each index of the string can be handled separately. For each i we need to check if s[i] can be transformed to t[i] using the mappings. Consider “a->b”, “b->c”, “c->d”, we can first transform a to b, then b to c and then c to d. So we transformed a to d. Many teams were not taking care of this issue.
Consider all the alphabet as nodes in a graph. Add an edge from node ‘a’ to node ‘b’ if mapping is “a->b”, now we can transform some char x to y if there is a path from x to y.
As the number of nodes is less (26), we can use floyd O(26^3) to solve it.
Link to solution: http://ideone.com/fak5Q3

** As I write this, it is 1 hour 40 minutes into the contest.
4 problems are cracked. And 3 teams are with 3 AC.

Cool Dude Laxman
Author: Anudeep Nekkanti
Tester: Mahbubul Hasan
Given N, K. Find number of arrays of size N, which are non-decreasing and product of any two numbers is <= K.
To the top teams! Why did it take so long? Why??
Array is non-decreasing, so <= K condition can be checked only for last 2 elements.
Now all you need is number of ways of arrays of size N, such that last 2 elements product is <= K.
Infinite if n = 1. K is 10^5, but last two elements product is <= K, so clearly last but one element can be at most sqrt(K).
DP[N][i] be number of ways to fill array of size N with last element being i. And array is non-decreasing.
Now loop over possible values of last but one element, and see what are valid for last. Add up.

Another version of the problem is to have N <= 10^9, K <= 10^5.
As solution is O(sqrt(K) * T). For this you need to notice that DP[N][i] can be calculated by using nCr()
Link to solution: http://ideone.com/e4Bs1I

Cracker shops
Author: Anudeep Nekkanti
Tester: Mahbubul Hasan
Let us break the problem into 2 parts.
First, for each i, find the least j such that j >= i and i..j covers all M crackers.
Second part, use this valid ranges to find the required answers.

First part is simple, you can use two pointer walk.
For second part, process from left to right, maintain a set of valid open ranges. Set top element should return the range of least length, upon ties it should return for least start index.
As you go you get answer for each of them.
One corner case is when every valid range ends at some p<k, then for all q>=k, we need to extend previous ranges.
Expected many more AC’s. It is also not too much to code once you get the idea.
Link to solution: http://ideone.com/v3SYp3

** 1 hour left in contest. 18 teams with 3 AC. I still have hopes that some team will get all 5 :)
Hurrey! IIT-D with 4 AC :-) !!! Awesome CMI with 4 AC :) I am winning Anup ;)

Jai Hanuman
Author: Anudeep Nekkanti
Tester: Mahbubul Hasan
* Each person has 3 chances for swaps. So he can get whatever he wants. But he might lose them later if someone else takes it from him.
* Each person gets his chance to ask for swap of crackers. This starts with person numbered 1, then 2, 3, 4 ... and ends with person numbered N.
Above line is the most important thing for the whole problem. See the chance happens in order. You got it?? There is no one to ask for swaps after N'th person. So he can always get whatever crackers he wants. No matter what happened before N'th persons turn, he will just swap to get whatever he wants.
* All N people are smart. They all make optimal choice. They all also know that others will always make optimal choice. Every person also knows about every other persons preference number (P[ ]).
N-1'th person knows that N'th will play smart and get whatever he wants. Best thing N-1'th person can do is to get whatever he wants without including what N'th person want.

Thats all, greedy solution. Loop from N to 1, for each position get the best possible according to P[]. To do this you can use segment trees or simple use set.
Link to solution: http://ideone.com/H8dB4N

** 45 minutes left, All 5 problems are cracked :D There are 3 teams with 4 AC.
** 5 minutes left, 3 teams got all 5 problems :D I won it :) :)

Let me tell you about what I really like about this set,
No solution needs a lot of coding or heavy data structures. All the solutions are simple loops and using build in data structures. 30-40 lines of code :)

Second thing I like about this set, I made the whole problem set (Ideas, solution, test data, statements) in 6 hours :)