

## Link to Contest:

<https://vjudge.net/contest/372067#overview>

Password: rccacm

## Problem A:

- Nearest Interesting Number

<https://codeforces.com/problemset/problem/1183/A>

- The input size for this problem is very small (only 1000), therefore we should be able to brute force the solution. I found the sum of the digits by first converting it into a string and then adding each digit - '0' to add its integer value. If the calculated sum is not divisible by 4, increase by one and try again.

## Problem B:

- Registration System

<https://codeforces.com/problemset/problem/4/C>

- This is a fundamental map problem. Use a map to keep track of the used strings and their frequencies. If the string is not in the map, then output "OK", otherwise output the string they entered and the frequency stored in the map. Check out the accepted code if you were unsure how to use the map.

### Problem C:

- Equalize Prices

<https://codeforces.com/problemset/problem/1183/B>

- The only values that matter for this problem are the cheapest and most expensive items in the array. I sorted it to find these values quickly. Add  $k$  (the given difference of prices) to the smallest value. If the biggest value in the array is  $2k$  bigger than the smallest, then it is impossible to make all the values equal (logically if you can add at most  $k$  to the smallest and subtract  $k$  from the biggest, it is not enough to make them equal). Otherwise the answer is the smallest value +  $k$ .

### Problem D:

- Computer Game

<https://codeforces.com/problemset/problem/1183/C>

- Due to the large input sizes, brute force will not work with this problem. First you want to see if the starting battery life is less than or equal to the number of turns in the game \*  $b$ . If it is, then it is impossible to complete the game. Otherwise the amount of battery we have left to spend just playing the game is **startingBattery - (turns \*  $b$ )**. Divide  $(a-b)$  from the remaining battery and then make sure that you take the minimum from that value and the number of turns in the game and you have your answer.

### Problem E:

- Candy Box

<https://codeforces.com/problemset/problem/1183/D>

- Begin the problem by inserting all of the array values into a map so that you have the keys followed by their respective frequencies. Then you want to store only their frequencies into a map or set (either works). Since we can only have unique numbers of candies for each gift, start with the type of candy with the highest frequency. If this frequency has not been used as a gift, add it to the answer. Otherwise subtract 1 until you find a frequency that has not been used yet.