

### SRM 172-2-1:

Time	Logging
12:41	<b>Start</b> reading
12:42	Trying to understand from the sample test cases
12:43	I think I'll create a vector of pair then sort it
12:44	No, It's not necessary
12:45	<b>Start</b> coding
12:45	I'll sort the vector to easily break ties after getting the result
12:48	Testing the code on all samples (kawigiEdit)
12:49	Passed and submitted with score 230.68
12:50	AC

Log Analysis
I didn't understand the problem fast enough.

### SRM 521-2-1:

Time	Logging
1:36	<b>Start</b> reading
1:38	<b>Start</b> thinking
1:39	doesn't seem to be greedy
1:40	Brute Force on all the possibilities and return the minimum ans
1:40	doesn't need to be verified and constraints are ok.
1:40	<b>Start</b> coding
1:42	Testing all samples
1:43	Passed all samples,submitted, WA!
1:44	if statement in the wrong position, the loops breaks earlier than I want.
1:45	AC

Log Analysis
I should've checked the corner cases before I submit.
I should've traced the code one more time before I submit.
Being in a hurry for a bigger score costed me a 10% penalty.

SRM 401-2-1

Time	Logging
------	---------

2:27	<b>Start</b> reading
2:28	Solved it before, remembered the idea.
2:28	<b>Start</b> coding
2:29	AC

Log Analysis	
None	

**SRM 268-2-2:**

Time	Logging
------	---------

10:56	<b>Start</b> reading
10:58	reading the samples to understand more
11:00	Brute Force will pass
11:00	<b>Start</b> coding
11:01	Insert in a set all concatenations of every 2 strings
11:02	Now, I can check correctly on the 1st condition of the problem but not the 2nd one
11:03	Push back the strings that are already there in the set in a vector of strings
11:03	Now I can check correctly on the 2nd condition
11:04	Testing all samples
11:04	Submitted, AC

Log Analysis
I didn't need to write a lot of logging since the problem was easy and clear.
Thinking phase was combined with the reading phase.
Unfortunately, there was an easier way to implement the code.

X

## SRM 362-2-2

Time	Logging
3:23	<b>Start</b> reading
3:24	read test cases to clarify the problem
3:25	<b>Start</b> thinking
3:28	Drawing squares
3:31	Test case 3 how?
3:33	Test case 3 outputs 9 not 14
3:35	Trying to find pattern from simpler cases
3:37	I have to understand test case 3
3:38	Drawing squares
3:41	Trying to find a pattern while ignoring test case 3 for a while
3:46	Stuck in test case 3
3:47	The squares have to be parallel to the axes
3:50	Drawing again
3:52	The combined squares don't count based on test case 2
3:53	The combined shape in test case 2 is a rectangle not a square!
3:54	I get test case 3 now
3:55	Now I can trace some numbers to find a pattern
3:59	Wrong drawing
4:03	Solved on paper from $n=4$ up to $n=16$
4:03	find pattern
4:08	can't find one yet

4:20	I think there might be more than 1 pattern
4:22	Still nothing
4:23	Hour is up, I'll see the editorial now
4:26	I didn't read it all, I received a hint to check if it's a perfect square
4:28	Idea started to be clear
4:32	If it's not a perfect square, how to solve that?
4:35	Editorial

Log Analysis
Never search greedily for a pattern while ignoring the actual problem
When test case 3's answer was 14 while I see it has to be 9, I had to know where to stop drawing different shapes of squares randomly. (My shaped drawing of squares was correct but I was not counting the combined squares).

## SRM 245-2-2

Time	Logging
9:09	<b>Start</b> reading
9:12	reading test cases to understand
9:17	Trying to understand
9:21	Still trying but there's some progress
9:24	How does he calculate the answer?
9:26	<b>Start</b> thinking
9:28	BF, but how to calculate the answer for all the conditions
9:29	If I figured it out, the problem is solved

9:35	Still trying using calculator
9:45	Ok, I think I got it.
9:48	<b>Start</b> coding
9:50	BF done, writing the equation now
9:51	Writing a function to simplify the code
9:55	Make everything a double
10:02	Testing passed
10:03	AC

Log Analysis
I have to be more organised while getting the idea (equation here).
I have to think in the equation depending mostly on the problem statement not on the test cases.

## SRM 496-2-2

Time	Logging	
12:55	<b>Start</b> reading	
12:55	reading test cases	
1:00	<b>Start</b> thinking	
1:02	Trying to figure out what to do	
1:03	Tracing an idea that I've got on the samples	
1:06	Having trouble with the last test case	
1:08	How is the answer equal to 4?	
1:11	aaaaaaaaaahhh :D	
1:11	My idea passed all samples now	

1:12	Trying to challenge my idea	
1:14	<b>Start</b> coding	
1:17	I don't need "flood fill" that much	
1:28	Testing,errors	
1:28	Tracing the idea	
1:30	Tracing the specific test case on the code	
1:33	Sample tests passed	
1:36	WA	
1:37	Back to thinking again (code is fine)	
1:39	Wait, I was doubting something	
1:40	I don't know how to verify it, I'll just try and submit it anyway	
1:43	WA again	
1:44	The problem is in the idea, now I'm sure	
1:45	Back to thinking	
1:47	another idea but I don't know if it's correct	
1:51	maybe calculating each one ('R' and 'B') independently would be better	
1:53	<b>Start</b> coding	
1:56	Instead of visiting, why not using just one (bool)	
1:59	AC	

Log Analysis
I found an idea, couldn't challenge it, so I dived in the idea ignoring the Brainstorm that could've saved me 20% penalty lost!
I think I had to wait and find other ideas and rank them.



## SRM 454-2-2

Time	Logging
5:30	<b>Start</b> reading
5:30	swap any two rows or any two columns (don't have to be neighbours)
5:32	Test cases
5:34	Just BF on all the strings and minimize?
5:34	<b>Start</b> thinking
5:35	Just that? (I mean my previous idea)
5:36	minimum number of swaps between 2 strings (I simplified the problem)
5:39	but how to do that?
5:41	giving myself some examples
5:42	Now, I know how to get a string from a string using swaps but I don't think that it will get me the minimum number of swaps
5:43	This seems easy but I'm not getting how to do it, why?!
5:49	I think I got it (check first if a single swap will help adjusting two positions at once and swap them, the remaining characters which are not at their positions should be swapped after that)
5:50	<b>Start</b> coding
5:59	Running samples
6:00	Got an error
6:00	Oh I forgot to return -1
6:00	Samples are okay now, AC

Log Analysis
I liked how the editorial described my way of thinking in this problem :D except for(the minimum number of swaps part)
I took advantage of the constraints to find the minimum number of swaps
I should've been faster

## SRM 408-2-2

Time	Logging
12:15	<b>Start</b> reading
12:15	Simulation problem?
12:17	Tracing samples
12:19	while loop and sort (decreasing) everytime
12:19	<b>Start</b> coding
12:21	Running samples
12:21	AC 3addany el 3eib :D

## SRM 563-2-2

Time	Logging
10:15	<b>Start</b> reading
10:17	exactly 1 coin out
10:17	dfs?
10:18	reading Test cases
10:18	Constraints are very small that I can do whatever I want
10:19	Maxsteps are only 10 steps

10:19	They just simulate the game (from test cases)
10:20	<b>Start</b> thinking
10:23	Just recursion and keep track numSteps when exactly 1 coin falls and minimize
10:24	<b>Start</b> coding
10:24	I have to be very careful when writing this kind of code
10:24	Get the position of each coin
10:25	Done, now write the dfs function
10:26	Parameters x1,y1,x2,y2,numSteps
10:27	make it void
10:28	bool valid function to check whether a coin is out or not
10:29	Use dx[] and dy[] for easier implementation
10:30	Start function with loop in 4 directions(just simulate the game)
10:32	Store the new position of the coins
10:33	What if the new position is a wall
10:33	just return to the old position
10:34	make sure the positions are valid before checking if it's a wall
10:34	check if exactly 1 coin is out
10:37	If so, then I reached the end of the game in this sequence of steps
10:38	so I should minimize numSteps with ans
10:40	oh, in "valid" function I should write if (x<0    y<0) not <=, fixed it
10:41	if both coins are still valid, continue the game and numSteps++
10:43	I think that's it
10:44	Testing to see how far I went
10:45	returns -1 only

10:46	<b>Start</b> debugging
10:48	oh no, I didn't call the function in "main"
10:49	errors in samples
10:50	I have to save the old position of the coins and assign it to them after I finish
10:55	I don't think I need any other parameters in the function other than numSteps
10:59	Compilation errors
11:11	Problems with eclipse
11:13	testing on the arena better
11:13	returns 1 only
11:36	Stuck

Log Analysis	
My solution was almost the same as the one in the editorial but I was tired and I wanted to solve this problem anyway so there was a lot of stupid mistakes in the code.	
In general, I need more practise on problems in which I use recursion	

## SRM 521-2-2

Time	Logging
11:14	<b>Start</b> reading
11:15	Trying to understand more from the sample test cases
11:15	Just erase all the matching parentheses and return the size of the string
11:16	<b>Start coding</b>
11:20	Running samples, error
11:21	I erase '(' before finding its ')'

11:22	AC
-------	----

Log Analysis
I could've avoided the debugging time completely If I concentrated more on the logic of the code while writing it.

## SRM-147-2-2

Time	Logging
2:06	<b>Start</b> reading
2:08	Ok, I understand
2:09	Reading test cases
2:11	<b>Start</b> thinking
2:11	string all of its characters are 'M' and then transform numFemales 'M' into 'F'
2:12	don't count 'F's while walking in the circle
2:13	I can't use pos+=k directly then
2:13	pos++ will pass
2:14	<b>Start</b> coding
2:18	calculate the new starting position
2:19	Assume the starting position is s.size()-1
2:19	Yes, that will work for every starting position correctly
2:20	AC

Log Analysis
I stopped coding for a while to think which was wrong

## SRM 427-2-2

Time	Logging
4:17	<b>Start</b> reading
4:20	I don't understand
4:22	Reading again
4:28	Not sure
4:28	<b>Start</b> thinking
4:29	calculate remainder of yearLength/dayLength and distribute it
4:30	how to do that while n can be $10^9$
4:30	Not $O(N)$ of course
4:33	verify remainder idea first on sample test cases
4:42	Trying to verify it without PC
4:45	Stuck at a test case
4:47	if dayLength is not divisible by (yearLength/dayLength ) is the problem
4:51	We're not searching for the remainder I guess
4:52	Try gcd
4:54	It works
4:54	<b>Start</b> coding
4:55	AC

Log Analysis
It was solved due to a pattern, while I should've tried to understand the problem more

because the pattern solution won't work every time.

## SRM 397-2-2

Time	Logging
3:30	<b>Start</b> reading
3:31	Test cases
3:34	<b>Start</b> thinking
3:35	iterate over all and maximize
3:36	Thinking about how to write the code
3:36	put 2 vectors in a vector of pair and sort before anything to break ties
3:37	After that, I iterate
3:37	<b>Start</b> coding
3:42	errors
3:42	<b>Start</b> debugging
3:43	if(i==j)continue; is wrong because I have to check that one too
3:44	AC

### Log Analysis

if(i==j)continue; I had to think about that earlier to avoid debugging

## SRM 530-2-2

Time	Logging
5:30	<b>Start</b> reading
5:35	every '.' in cutter must be placed on 'X' of cake
5:35	<b>Start</b> thinking
5:37	Thinking pictorially by looking at the illustration of test case 0
5:39	Since we're simulating backwards, then every '.' must be on '.'
5:41	I have a BF simulation idea
5:42	Try to put the cutter on every cell of the cake. If it doesn't fit or doesn't fulfill the conditions continue;
5:43	else simulate the game, if '.' is placed on '.' make it 'X'
5:44	After finishing check if all the cells of cake are 'X'
5:45	I will verify this idea on test case 0
5:48	well, it works and the constraints are definitely ok.
5:50	<b>Start</b> sketching the code
5:50	2 for loops on cake, if cutter won't fit continue;
5:53	cutter's coordinates are different from cake's coordinates
5:57	Got it, cake's coordinates are (i+a)(j+b)
5:57	<b>Start</b> coding
6:04	<b>Start</b> debugging
6:04	remove the continue part and adjust the conditions of the for loops. That would be better.



6:07	Stupid mistake, cutter[i] while it should've been cutter[0]
6:08	Submit
6:08	WA
6:09	The same mistake was written in somewhere else in the code -_-
6:09	AC

Log Analysis
When discovering a bug in the code, I should search for a similar one in the rest of the code.

## SRM 430-2-2

Time	Logging
9:36	<b>Start</b> reading
9:37	<b>Start</b> thinking
9:38	the only difference between “or” and “+” is $1+1$ (1 1)
9:39	under each ‘1’ of x in binary there has to be a ‘0’
9:41	under each ‘0’ of x in binary we can put a ‘0’ or a ‘1’
9:43	but how can I find the kth number?
9:43	follow the pattern from the smallest number and see when he puts ‘0’ and when he puts ‘1’
9:46	I need a number $> 5$ because 5 contains only one ‘0’
9:50	I know the pattern now but I can’t simulate it because k can be too big
9:54	maybe an equation?
9:59	Try to find a pattern
10:02	lol, there’s a pattern between k and the answer
10:03	putting k(in binary) at the empty slots in y gives the answer
10:03	<b>Start</b> code sketching
10:04	convert x and k to binary and put them into 2 strings a and b
10:03	if(a[i]=='1')a[i]='0';
10:05	else a[i]=b[b.size()-1] and erase last char in b
10:06	After finishing, if b isn’t empty, append b to the beginning of a
10:07	convert a to decimal and put the answer in long long

10:07	<b>Start</b> coding
10:11	AC

Log Analysis
Writing the code was a lot easier after figuring out what I'm going to do in the "Code Sketching" phase.

## SRM 432-2-2

Time	Logging
8:28	<b>Start</b> reading
8:29	<b>Start</b> thinking
8:30	while(k--) flip the column with most zeros?
8:32	Try this idea on test 0 and make $k > 1$
8:33	Observation: flipping a column twice makes it return to its initial condition
8:34	keep this idea and think of another one. Not sure if I can verify my idea.
8:35	Concentrate on the rows now since I can't find any other idea on the columns
8:37	Try bigger K and see what happens
8:42	equal rows stay equal no matter what
8:45	the max number of equal rows is the answer(if no flipping or no lit)
8:46	If we can play on the most frequent row and make it lit, then its frequency is ans.
8:47	Turn all zeros to ones. if(zeros > k) this row cannot be lit.
8:47	if(zeros == k) then we can turn all zeros to ones
8:48	what if $k > \text{zeros}$ ?
8:49	turn all zeros to ones and I will have some remaining flips( $k - \text{zeros}$ )
8:50	I can use them on just one column and if $(k - \text{zeros}) \% 2 == 0$ then it's ok
8:50	<b>Start</b> code sketching
8:54	<b>Start</b> coding
8:59	AC

Log Analysis
More time on code sketching would've been better

## SRM 376-2-2

Time	Logging
8:06	<b>Start</b> reading
8:08	Test cases
8:12	<b>Start</b> thinking
8:14	Iterate over all cells and check if it's reachable by the shortest path
8:16	how to handle 'I' , '-' and '+' ?
8:21	keep track of old position of (x,y) and compare it with the new one.
8:25	don't iterate over all cells, BFS would do it.
8:27	Simulate the process using if conditions to handle 'I' , '-' and '+'
8:28	Store in each cell the amount of fuel needed to reach it in another 2D arr, that way I can use a cell's amount to find another cell's amount too.
8:28	<b>Start</b> code sketching
8:35	in dx[] and dy[] make the first two up and down and the other two left and right to make the if conditions easier
8:40	<b>Start</b> coding
8:53	AC

Log Analysis
I spent more time on coding than the time I spent on code sketching.

## SRM 470-2-2

Time	Logging
9:13	<b>Start</b> reading
9:23	<b>Start</b> thinking
9:24	insert a character that represents the trophy room
9:26	verifying where to insert
9:29	inserting in the position mentioned will work under all conditions
9:30	divide the string into two parts (that's better)
9:32	in each turn simulate the game exactly like in the problem statement
9:34	for each turn, the players chooses the most frequent color in his part of the string
9:35	No, challenged it.
9:38	frequency=the most frequent color in player 1's part of the string if(frequency of color c in string1>frequency of color c in string 2) choose it
9:41	storing all frequencies in s1 and pair them with their (f1-f2) f1 freq in string 1, f2 freq of the same color string 2
9:46	Challenged it
9:57	found an idea and can't challenge it yet
10:03	<b>Start</b> code sketching
10:14	<b>Start</b> coding
10:19	AC

### Log Analysis

Challenging your idea is great.

## SRM 369-2-2 (TWINS)

Time	Logging
4:24	<b>Start</b> reading
4:26	reading test cases
4:28	<b>Start</b> thinking
4:29	can we simulate it?
4:34	challenged an idea that we've got
4:35	make all the string 'A's and distribute 'B' all over it
4:50	order 4 million operations max
4:51	verify on samples
4:56	<b>Start</b> code sketching
5:22	<b>Start</b> coding
5:31	<b>Start</b> debugging
5:34	Submitted, WA
5:36	ma3'reb prayer
7:18	<b>Start</b> thinking
7:26	thinking symbolically
8:11	<b>Start</b> coding
8:15	<b>Start</b> debugging
8:18	AC

Log Analysis
Diving into our first idea till we got WA was very wrong, and brainstorming was the right thing to do.

## SRM 383-2-2

Time	Logging
4:24	<b>Start</b> reading
4:28	<b>Start</b> thinking
4:29	Try all the lengths from 1 to maxLength
4:34	if cutting something into that certain length is so expensive don't take it
4:35	<b>Start</b> code sketching
4:50	<b>Start</b> coding
4:51	error in last test case
4:56	<b>Start</b> debugging
5:22	tracing this case on the code
5:31	problems in my idea
5:34	<i>I calculate the number of cuts wrong because I miscalculated it while understanding the test cases in the first place</i>
5:36	AC

Log Analysis
More focus in the beginning would've saved me 14.867 debugging minutes.



## SRM 419-2-2

Time	Logging
7:46	<b>Start</b> reading
7:47	wow, in “undo t” t can be $10^9$
7:48	I don’t think it matters
7:49	<b>Start</b> thinking
7:49	I can just simulate it
7:50	when I find undo I go back to the beginning of the vector and undo each thing in the “undo” seconds time.
7:51	verify it
7:53	find a better idea
7:53	try iterating backwards
7:55	it works on samples
7:56	<b>Start</b> code sketching
8:03	<b>Start</b> coding
8:06	WA
8:08	instead of writing <code>v[i].substr(5,v[i].size()-5)</code> ; I wrote <code>v[i].substr(5,v.size())</code> ; <code>--</code>
8:10	AC

Log Analysis

If my stupid mistake happened in a SRM, it would be catastrophic! Maybe I did it because I'm not very focused now but I'll put it in the errors inspection list just in case.

## SRM 565-2-2

Time	Logging
5:42	<b>Start</b> reading
5:44	I'm thinking recursion
5:44	max 20 elements but each element can be 20,000,000
5:45	Test cases
5:45	<b>Start</b> thinking
5:45	Recursion, try to take it and try not to take it
5:46	make it long long because of the limits
5:46	<b>Start</b> code sketching
5:48	<b>Start</b> coding
5:51	when I make <code>vector&lt;long long&gt;</code> = to another <code>vector&lt;int&gt;</code> it doesn't work but I realised I didn't need it
5:52	AC

Log Analysis
I didn't need to make the <code>vector&lt;long long&gt;</code> , <code>vector&lt;int&gt;</code> was good. I should've realised that in the code sketching time.

## SRM 414-2-2:

Time	Logging
3:31	<b>Start</b> reading
3:37	<b>Start</b> thinking
3:39	I have to check on all the starting times
3:40	Brute force but how?
3:44	Iterate over every hour and minimize the answer but how to calculate the answer
3:56	<b>Start</b> code sketching
4:01	<b>Start</b> coding
4:06	AC

Log Analysis
I had to be a lot more quicker in calculating the answer.

**SRM 431-2-2:**

Time	Logging
4:26	<b>Start</b> reading
4:27	<b>Start</b> thinking
4:27	First point always falls till $y=0$
4:28	How to get $y$ from distance using the distance equation
4:30	Got an equation to calculate $y$
4:30	<b>Start</b> code sketching
4:33	<b>Start</b> coding
4:36	AC

Log Analysis
According to the target time (0.3 minutes), I wasn't quick enough.

**SRM 492-2-2:**

Time	Logging
5:36	<b>Start</b> reading
5:39	<b>Start</b> thinking
5:40	How to know that 2 points lie on a single straight line?
5:40	Try to know from samples
5:44	Calculate the slope to know
5:45	How to know the minimum number of trees that need to be trimmed down?
5:46	Trace another sample using the slope
5:50	BF on all possible slopes?
5:53	Am I calculating the slope correctly?
5:54	Yes, I am I am.
5:57	Thinking symbolically
6:02	Try the slope between every 2 points on all the other points

6:04	Testing that on samples
6:06	No, find a better idea
6:14	The line will be decreasing, increasing or horizontal
6:22	Try the slope between every 2 points, and loop on all the other points while minimizing
6:25	while looping calculate y using the slope equation and $cnt++$ if $y < y[i]$
6:27	<b>Start</b> code sketching
6:28	Watch out for double precision and use EPS
6:30	<b>Start</b> coding
6:37	<b>Start</b> debugging
6:42	a/b while it should be b/a
6:44	AC

Log Analysis
Not code sketching correctly and being in a rush costed me debugging time

## SRM 498-2-2

Time	Logging
8:16	<b>Start</b> reading
8:19	<b>Start</b> thinking
8:19	5 nested loops? :D
8:20	Actually, that might work
8:20	All possible values of a,b,c&d and check. (That's 4 nested loops then)
8:21	<b>Start</b> code sketching
8:28	<b>Start</b> coding
8:36	Errors in samples
8:36	<b>Start</b> debugging
8:36	Trace the test where it fails
8:38	Submitted, WA
8:41	AC

Log Analysis
I was writing <code>int n=seq.size()-1</code> instead of <code>int n=seq.size();</code> ; so the WA doesn't count :D

## SRM 350-2-2

Time	Logging
8:13	<b>Start</b> reading
8:14	<b>Start</b> thinking
8:16	generate numbers $\leq 5000000$ that are perfect powers
8:18	Think of an idea that has an appropriate order
8:27	Thinking of a greedy idea
8:36	I think sieve would do it
8:40	When doing sieve, we mark only numbers that their sum+0 equals a desired number
8:40	Handle the rest with a loop
8:41	<b>Start</b> code sketching
8:45	<b>Start</b> coding
8:53	<b>Start</b> debugging
8:56	is the problem in eclipse?



8:57	I don't know what's wrong
9:02	Test it on the arena, segmentation fault
9:05	AC

Log Analysis
When I tested my code on the arena instead of eclipse, it gave me "segmentation fault". That made me realise quickly where my mistake was. I should do that later.
I should've considered that when multiplying x by i and then writing arr[x], x might be out of boundaries.

## SRM 402-2-2

Time	Logging
1:15	<b>Start</b> reading
1:17	<b>Start</b> thinking
1:17	Greedy, check if all the integers exist then return a vector of $2(v[0]-1, v[v.size()+1])$
1:18	If no, return the missing number, if more than one number is missing, return empty vector
1:18	How to do so while each number is from 1 to $10^9$
1:21	vector<int> differences, if all=1 return smallest-1, biggest+1
1:22	If no, check if all differences =1 except 1 diff that is ==2 and return the missing, if no return empty vector
1:22	<b>Start</b> code sketching
1:26	<b>Start</b> coding
1:30	Wrong test case because vector<int> ans can't contain a zero, fixed it
1:31	AC

### Log Analysis

I didn't notice that it has to be all positive. If the sample tests didn't contain such test case, I could've got a WA.

## SRM 364-2-2

Time	Logging
2:23	<b>Start</b> reading
2:26	Reading samples
2:27	implementation problem
2:28	<b>Start</b> thinking
2:28	It's just a coding problem
2:28	How to code it?
2:29	map each player with its score
2:29	2D array containing team names and in each team it contains its players
2:30	Check if both players are from the same team when they hit each other
2:30	<b>Start</b> code sketching
2:36	Now, I have each team with its players in a 2D array
2:44	How to sort lexicographically

2:52	<b>Start</b> coding
3:10	Compilation errors
3:10	I didn't know that I can't use <code>max_element</code> with a vector of pairs
3:10	<b>Start</b> debugging
3:11	How to sort lexico... without <code>max_element</code>
3:17	Done, but there's one more compilation error that I don't understand
3:20	ok fixed it, but now errors in sample tests
3:21	Trace last test case
3:21	Make it test case 3 because it's shorter
3:26	I was writing <code>z.size()</code> instead of <code>z[i].size()</code>
3:26	WA
3:27	checking the code again
3:32	read the problem statement again
3:34	can't find what's wrong
3:34	go back to reading the code
3:42	can't find any mistake
3:49	check the test case that gives WA
4:12	AC

Log Analysis
<p>OK, first the compilation errors due to lack of C++ knowledge.</p> <p>Second, I was writing <code>tmp</code> instead of <code>tmp[i]</code></p> <p>Third, I was writing <code>z.size()</code> instead of <code>z[i].size()</code></p> <p>After fixing all of this, I got WA.</p> <p>I took a long time searching for my mistake and I didn't want to see the editorial because I was very convinced that my code is the right solution.</p> <p>The mistake was that I'm using <code>find(all(z))</code> instead of <code>find(z.begin()+1,z.end())</code> since I didn't</p>

want to search in the first element.

I wrote the proper solution and my idea was correct. But a few mistakes happened due to lack of C++ knowledge and due to (ma5adtresh baly) since the code wasn't short . I wrote my mistake in the errors inspection list hoping that these kinds of mistakes don't happen again.

## SRM 355-2-2

Time	Logging
8:02	<b>Start</b> reading
8:03	<b>Start</b> thinking
8:04	Well, if one of the 2 numbers doesn't contain 8 return 0; else we have to check
8:06	the last part of the number is the changing part while increasing
8:06	I have an idea to check the first number's first eights if they were a prefix of the second number
8:06	It works on all samples
8:07	Challenge it
8:07	Challenged it but I don't think it's wrong. It needs improvement.
8:10	Thinking with examples.
8:12	If !same sizes return 0;
8:14	erase characters from begining that are !=8

8:18	<b>Start</b> code sketching
8:21	<b>Start</b> coding
8:25	AC

Log Analysis
I got AC but I didn't verify the idea quite enough. In a real SRM, I would've stayed longer in verifying which I should've done while practising.

## SRM 417-2-2

Time	Logging
10:17	<b>Start</b> reading
10:21	<b>Start</b> thinking
10:21	Generate all substrings and put them in a vector and sort them
10:23	if two have same score take the one with lower prefix, how to handle that?
10:23	Ok, got it
10:24	<b>Start</b> code sketching
10:27	<b>Start</b> coding
10:35	AC

Log Analysis
I had to be faster

## SRM 472-2-2

Time	Logging
12:14	<b>Start</b> reading
12:16	<b>Start</b> thinking
12:17	$4^{14}$ is the max power that can be used
12:18	how is playing optimally?
12:20	There's more than one scenario in a game
12:24	It's not possible to use recursion while $n$ can be $10^9$
12:29	Thinking greedy
12:41	There's a pattern for the times when Hanako wins
12:41	increases +2 and then +3 and then +2 and so on
12:41	Transform this into an equation
12:45	All multiples of 5 are included, still trying to get the equation
12:46	Since it's increasing by 2 after a multiple of 5, then $n \% 5 == 2$

12:47	Ok, I got the equation
12:48	I'll just try this solution
12:49	Start coding
12:50	AC

Log Analysis
I don't know what to write :D

## SRM 425-2-2

Time	Logging
2:13	<b>Start</b> reading
2:18	<b>Start</b> thinking
2:22	I don't know how to start
2:26	I can't solve it using only an equation, let's think recursion
2:30	I can easily simulate the robot's movements using recursion but how can I calculate the probability while doing it?
2:39	can't figure it out yet
2:47	maybe summing the probabilities would do it
2:48	I don't have anything else to do, so I'll just try it since I can't verify it on paper
2:48	<b>Start</b> code sketching
2:50	<b>Start</b> coding
2:55	AC

Log Analysis
It took me a lot to know how to calculate the probability

## SRM 416-2-2

Time	Logging
4:10	<b>Start</b> reading
4:11	<b>Start</b> thinking
4:13	The biggest value of $n(10^9)$ is represented in only 30 digits in binary
4:16	The solution will come from only working on binary values but how?
4:21	Observing what happens when we increase a binary value
4:22	we can transform $n$ to binary and make next_permutation and check on each permutation if it's bigger than $n$ , but it won't pass the time
4:23	Actually the first next_permutation will always be bigger than $n$
4:24	Verify on samples
4:25	The solution can have more digits than $n$
4:25	Put leading zeros in $n$ 's binary value but how many?
4:26	I think only one would be enough



4:26	No need for code sketch
4:26	<b>Start</b> coding
4:31	

Log Analysis
Is it ok to ignore code sketching if the code is in your mind and it's easy?

## SRM 526-2-2

Time	Logging
9:12	<b>Start</b> reading
9:15	<b>Start</b> thinking
9:15	The first thing that pops into my head is Brute Force
9:16	Try each row, each column and the return min
9:16	verify quickly on samples
9:17	It's all about how to write the code now
9:18	<b>Start</b> code sketching
9:20	Ok, I checked on rows. Now, I'll check on columns.
9:25	I think there's a coding mistake. I'm trying to fix it.
9:30	<b>Start</b> coding
9:33	errors in test cases

9:33	<b>Start</b> debugging
9:33	Trace test case 1
9:40	Fixed the columns, now I'll fix the rows
9:48	No!!! I didn't think about something important in the problem!
9:51	Back to thinking
10:08	I need transition
10:26	AC

Log Analysis
I think I had the chance to solve this problem on my own if I noticed that part that I missed.

## SRM 568-2-2

Time	Logging
1:01	<b>Start</b> reading
1:02	<b>Start</b> thinking
1:03	I'm thinking brute force
1:04	3 nested loops for the colors and one for calculating the moves
1:05	I can't verify it on paper but I think it's correct
1:05	<b>Start</b> coding
1:11	AC

## SRM 372-2-2

Time	Logging
1:42	<b>Start</b> reading
1:54	<b>Start</b> thinking
1:57	I don't have any other idea than simulating it
1:59	How can I simulate it?
2:02	The car with the lowest number lane that yielded exists first.
2:13	Trying to think how to implement the code
2:15	<b>Start</b> code sketching
2:16	use indices instead of erasing to avoid mistakes
2:23	<b>Start</b> coding
2:26	AC

## Log Analysis

I should've wrote more log

## SRM 526-2-2

Time	Logging
1:22	<b>Start</b> reading
1:25	<b>Start</b> thinking
1:25	I'm thinking on simulating it
1:26	I can't because T can be $10^9$
1:28	I think I can solve it by observing what happens when I put the clipboard
1:29	Observing the illustrated example in the samples
1:35	Thinking pictorially
1:45	There's some kind of pattern
1:54	verifying an equation to solve the problem on samples
1:58	The idea works
1:58	Try to challenge it
2:02	Challenged it

2:03	How to improve my idea or use it to come up with another one
2:12	I'll iterate to 100 so that my code passes my test case
2:14	<b>Start</b> code sketching
2:15	watch out for overflow
2:17	<b>Start</b> coding
2:23	error in last test case
2:23	<b>Start</b> debugging
2:26	AC

Log Analysis: It was nice that I challenged my idea but I hope to be faster in thinking by practising more.

### **SRM 373-2-2**

Time	Logging
12:49	<b>Start</b> reading
12:52	understanding sample test cases
12:55	<b>Start</b> thinking
12:56	BF but how?
1:07	I'll try putting the max possible number of words on each line
1:10	How to implement the code?
1:15	I think it's better to code sketch
1:15	<b>Start</b> code sketching
1:29	<b>Start</b> coding
1:38	error in some test
1:38	<b>Start</b> debugging
1:41	AC

Log Analysis
I had to be a lot faster in code sketching.

## SRM 597-2-2

Time	Logging
1:18	<b>Start</b> reading
1:20	<b>Start</b> thinking
1:20	Observe what happens in examples
1:20	verify on samples
1:36	It works
1:37	<b>Start</b> code sketching
1:40	<b>Start</b> coding
1:43	AC

Log Analysis
I had to write more log. It's just that I was busy in observing what happens in examples.

## SRM 390-2-2

Time	Logging
12:27	<b>Start</b> reading
12:28	<b>Start</b> thinking
12:33	concatenate while using mode so the number doesn't get too big
12:36	ans is equal to -1 if the same number comes up again after concatenating using mod
12:37	handle that with a map to check visited
12:37	<b>Start</b> code sketching
12:41	<b>Start</b> coding
12:47	error in last sample
12:47	<b>Start</b> debugging
12:49	AC

Log Analysis
I was getting an error in a sample test, I changed something in my code and it passed all samples then got AC. I don't know why my solution worked which is bad.

## SRM 403-2-2

Time	Logging
1:12	<b>Start</b> reading
1:13	<b>Start</b> thinking
1:16	push back in a vector all lucky numbers of all the sizes between a and b
1:16	how to generate lucky numbers needs code sketching
1:17	<b>Start</b> code sketching
1:24	<b>Start</b> coding
1:29	Testing my code on the arena with the worst case
1:29	AC

Log Analysis
It was an easy problem and I could've solved it faster.



## SRM 576-2-2

Time	Logging
6:25	<b>Start</b> reading
6:29	Test cases
6:33	<b>Start</b> thinking
6:34	We don't need to call anything horizontally, everything has to be calculated if vertical
6:36	I think it's about doing dfs but with a little optimization
6:40	I'll write normal dfs but count only when going up or down
6:42	Ok, that's the idea, all I need now is to sketch it
6:43	<b>Start</b> code sketching
6:48	I'm not sure if my sketched code will work, but I'll go ahead and try it.
6:48	<b>Start</b> coding
6:57	<b>Start</b> debugging

7:12	my idea of counting when moving vertically only works, the problem is in shortest path and backtracking
7:46	Transition
7:53	AC

Log Analysis
The problem is that I sketched the code and I wasn't sure that it will perform what I had in mind.

## SRM 490-2-2

Time	Logging
9:45	<b>Start</b> reading
9:49	understanding how the answer is calculated
9:52	I can't get it yet
9:55	I'll consider understanding the formula as a part of thinking
9:56	<b>Start</b> thinking
10:05	thinking in other things than calculating the answer now
10:29	I don't know how to approach this problem
10:32	I'm trying to find a relation between anything but nothing
10:42	I'm lost
10:59	Transition
11:17	AC

Log Analysis
I didn't know how to think in an organised way to solve this kind of problems

## SRM 415-2-2

Time	Logging
11:52	<b>Start</b> reading
11:58	can't understand the coordinates part yet
12:00	<b>Start</b> thinking
12:00	Brute Force I think
12:01	I think I can make 3 nested loops to check but how can I check if !valid
12:07	I think I got it
12:07	<b>Start</b> code sketching
12:14	<b>Start</b> coding
12:19	WA
12:21	<b>Start</b> debugging
12:26	AC

Log Analysis
I should've verified my idea in return -1 first to save myself the 10% penalty.

## SRM 591-2-2

Time	Logging
12:04	<b>Start</b> reading
12:07	<b>Start</b> thinking
12:12	I remember solving it before
12:14	I'll just start coding without sketch
12:15	<b>Start</b> coding
12:17	AC

## SRM 429-2-2

Time	Logging
1:45	<b>Start</b> reading
1:51	<b>Start</b> thinking
1:51	I solved it before but still thinking
1:52	get the equation to calculate the number of cells from test case 1
2:00	I think I got it
2:00	some problems with kawigi
2:04	<b>Start</b> coding
2:06	done coding
2:07	copying the code to the arena without kawigi now
2:10	<b>Start</b> debugging
2:13	AC

### Log Analysis

It's very important to think after solving a hard problem how you're going to solve it the next time you see it and how you will avoid repeating the same mistakes.

## SRM 532-2-2

Time	Logging
2:51	<b>Start</b> reading
2:56	<b>Start</b> thinking
2:57	thinking greedy
2:58	make the most beautiful one in the middle
3:01	I'm getting closer
3:01	make 3 vectors left,right and middle
3:03	sometimes it can be right and left
3:03	put it right or left according to the greater value
3:04	<b>Start</b> code sketching
3:09	<b>Start</b> coding
3:18	errors in tests

3:19	<b>Start</b> debugging
3:22	Trace
3:25	my idea is correct in this specific case, there's a problem in the code then (Although, the idea should be modified)
3:29	fixed that case but still error in last test case
3:29	now fix the flaw I found in the idea
3:30	Back to thinking
3:43	The problem is only concerning one case in the problem
4:00	<b>Start</b> coding
4:06	<b>Start</b> debugging
4:13	YESS!! AC

**Log Analysis:** I shouldn't assume things greedily without verifying them like that.

## SRM 365-2-2

Time	Logging
8:11	<b>Start</b> reading
8:19	Trying to understand
8:27	<b>Start</b> thinking
8:29	try to get the number of divisors of $n$ that gives rem of 1 and rem of 3 when divided by 4
8:37	we want to optimize a $2 \cdot 10^9$
8:50	make $\text{sqrt}(n)$
9:00	optimizing
9:11	<b>Start</b> code sketching
9:15	<b>Start</b> coding
9:20	AC

## SRM 578-2-2

Time	Logging
8:16	<b>Start</b> reading
8:18	test cases
8:19	<b>Start</b> thinking
8:20	I understand the problem but how to calculate the # of all the possible sets?
8:30	thinking on the samples
8:34	of course, we can't calculate the ans using cnt++ only. There has to be some equations.
8:40	Observing
8:44	I noticed earlier in 1st test the fact that $\text{ans} = 2^3 - 1$ and now I think it's important
8:45	I should get x such that $\text{ans} = 2^x - 1$
8:48	I think x is # of connected components but within dist



8:48	verify
8:50	<b>Start</b> code sketching
8:55	<b>Start</b> coding
9:00	<b>Start</b> debugging
9:02	AC

Log Analysis
I hope I can be faster next time in figuring out the idea because these problems are mostly coding (not thinking) problems I guess.

## SRM 548-2-2

Time	Logging
10:08	<b>Start</b> reading
10:11	<b>Start</b> thinking
10:12	Observing what happens
10:24	I need BF
10:32	how can I optimize my BF?
10:35	Binary search would work here
10:35	<b>Start</b> code sketching
10:38	<b>Start</b> coding
10:41	AC

Log Analysis
My thinking has to be straight and fast so I can get a higher score in Div1 250

## SRM 553-2-2

Time	Logging
8:24	<b>Start</b> reading
8:29	<b>Start</b> thinking
8:30	observing what happens in test cases
8:32	simulate it?
8:33	trying simulation on test cases
8:38	simulation works but I need more cases and I also need some tricky ones.
8:40	challenging my self
8:43	my idea isn't complete at all, but giving myself more tests helps
8:49	I think my idea is ok now.
8:49	<b>Start</b> code sketching

9:01	<b>Start</b> coding
9:13	<b>Start</b> debugging
9:20	WA
9:38	Trying to implement a part of the idea but facing some problems
9:41	WA again
9:55	I had <code>int x=0;</code> and it should've been long long like the vector!
9:56	WA again
9:59	found some other problem, fixing it
10:10	AC

Log Analysis
I should've been more careful while code sketching/coding. I should've tried to fix all my mistakes at once to reduce the number of these WAs to one.

## SRM 438-2-2

Time	Logging
11:55	<b>Start</b> reading
12:00	<b>Start</b> thinking
12:00	They are only 2 elements. Can I just try all the possibilities?
12:03	<b>Start</b> code sketching
12:11	there's a test case that I didn't understand correctly
12:14	got it
12:14	continue sketching
12:19	<b>Start</b> coding
12:31	<b>Start</b> debugging
12:36	fixed an error in the euclidean distance function

12:45	WA
12:57	AC

Log Analysis
I didn't want to think more in how to implement the code using nested loops and I rushed into coding and implementing it manually.
In the code sketch phase, I discovered that I didn't understand a test case very good.

## SRM 556-2-2

Time	Logging
8:47	<b>Start</b> reading
8:52	<b>Start</b> thinking
8:54	Obviously, it's a DFS/BFS problem.
8:56	I can solve it recursively
8:56	I can use some memoization too
9:00	<b>Start</b> code sketching
9:05	<b>Start</b> coding
9:10	<b>Start</b> debugging
9:15	AC

## SRM 495-2-2

Time	Logging
12:21	<b>Start</b> reading
12:24	<b>Start</b> thinking
12:25	first do sieve, so we can have all primes ready
12:26	observing test cases
12:31	I think I got it
12:33	organising the idea in my mind
12:34	<b>Start</b> code sketching
12:47	<b>Start</b> coding
12:56	<b>Start</b> debugging
2:05 (t2riban)	AC

### Log Analysis

I didn't try to think in the best way to solve this problem. I found a BF solution that could pass and I dived into it. It wasn't easy to implement so it costed me a lot of debugging time.

### SRM 366-2-2

Time	Logging
8:47	<b>Start</b> reading
8:52	recursion
8:54	<b>Start</b> thinking
8:56	(garraab marra add w marra subtract)+memoization
9:05	<b>Start</b> coding
9:15	AC

## SRM 412-2-2

Time	Logging
11:44	<b>Start</b> reading
11:58	<b>Start</b> thinking
11:58	It's about how can I implement the code
11:59	I think I'll waste my time in sketching the code. It will also be hard.
12:00	<b>Start</b> coding
12:45	<b>Start</b> debugging
1:11	no idea why my sort function doesn't work correctly

Log Analysis
I was using <code>sort(v.begin(),v.end(),compare);</code> for the second time. I had a problem in the

"compare" function.
I also wasn't in a good condition while I was solving this problem.
I wasn't confident about the functionality of my code while writing it which made me count some stuff to make sure that it will work.

## SRM 487-2-2

Time	Logging
7:37	<b>Start</b> reading
7:43	<b>Start</b> thinking
7:43	1st thing came to my head is to try all the possibilities
7:44	rec+dp or what?
7:46	how can I use rec in this problem?
7:46	should I use something else?
7:50	I'm thinking in some BF way and fakes recursion
7:52	avoid all the pairs then avoid recursion altogether?
7:57	observing test cases
8:01	I think original rec that I thought of at 1st was the best thing to do



8:06	It'll be easier to get all the pairs then do rec on them
8:09	think in other way than rec
8:17	I have another idea but still organising it
8:19	got it
8:19	<b>Start</b> code sketching
8:22	<b>Start</b> coding
8:24	<b>Start</b> debugging
8:25	AC

Log Analysis
rabena y5alelna el brainstorm :D

## SRM 421-2-2

Time	Logging
10:15	<b>Start</b> reading
10:17	<b>Start</b> thinking
10:22	I need to search between each 2 points for P
10:27	I think I can use BS instead of iterating between each 2 points
10:31	<b>Start</b> code sketching
10:40	<b>Start</b> coding
10:47	AC

## SRM 444-2-2

Time	Logging
12:34	<b>Start</b> reading
12:37	<b>Start</b> thinking
12:37	big limits, it's a math problem
12:37	factorize?
12:39	observing what happens when factorizing
12:46	verifying some idea on samples
12:47	I'll go ahead and code it
12:47	<b>Start</b> coding
12:49	done, but arena is disconnected
tany yom	AC(arena didn't want to open sa3etha)

## SRM 484-2-2

Time	Logging
6:13	<b>Start</b> reading
6:14	<b>Start</b> thinking
6:16	observing test cases
6:19	thinking about rabbit numbers
6:29	I'm stuck,I'll use eclipse to observe rabbit numbers
6:33	cout some rabbit numbers, trying to observe
6:34	They contain only 0,1,2,3
6:35	The list of rabbit numbers is generated like binary numbers somehow
6:36	how to compose this sequence
6:43	I have a stupid idea but it'll work

6:44	<b>Start</b> coding
6:54	<b>Start</b> debugging
	AC

Log Analysis
I had to start observing on eclipse earlier

## SRM 411-2-2

Time	Logging
1:19	<b>Start</b> reading
1:22	<b>Start</b> thinking
1:22	we'll have to try them all
1:25	I think recursion
1:27	but how to use it
1:32	<b>Start</b> code sketching
1:39	<b>Start</b> coding
1:49	<b>Start</b> debugging
1:57	AC

Log Analysis
I need more practise on dp problems

## SRM 567-2-2

Time	Logging
3:19	<b>Start</b> reading
3:20	<b>Start</b> thinking
3:20	let's simplify this equation first.
3:21	simplified to $a + 2\sqrt{a \cdot b} + b$
3:22	what matters is $\sqrt{a \cdot b}$
3:24	in other words, if $(a \cdot b)$ is a perfect square then ok.
3:26	how to check on all pairs of $a, b$
4:00	stuck
4:07	editorial

Log Analysis
I don't know if I could reach such observations on my own to solve this problem.

## SRM 354-2-2

Time	Logging
6:26	<b>Start</b> reading
6:29	<b>Start</b> thinking
6:32	take the smallest date that is possible
6:33	we have to check if it's possible 1st
6:36	<b>Start</b> code sketching
6:41	<b>Start</b> coding
6:51	<b>Start</b> debugging
6:52	AC

## SRM 357-2-2

Time	Logging
8:57	<b>Start</b> reading
8:58	max constraint is 20=use recursion
8:59	reading some test cases
9:00	<b>Start</b> thinking
9:00	obviously, I'll try all possibilities and minimize. But how?
9:06	<b>Start</b> code sketching
9:08	<b>Start</b> coding
9:11	AC

## Log Analysis

I cycled during coding for setting memo[] with a smaller size.

## SRM 523-2-2

Time	Logging
12:44	<b>Start</b> reading
12:46	<b>Start</b> thinking
12:47	math akid
12:50	we need to count arithmetic alone and geometric alone
12:57	geometric sequence can be simulated
12:58	let's do problem simplification and calculate the answer to arithmetic only
1:04	I think I got the proper equation for that now
1:05	now,simulate geometric sequence and check for the common numbers
1:07	<b>Start</b> sketching
1:10	<b>Start</b> coding



1:13	AC
------	----

Log Analysis
Problem simplification 7elw :D

## SRM 413-2-2

Time	Logging
10:19	<b>Start</b> reading
10:22	reading test cases to understand more
10:24	I read a part again in the problem statement that made me understand now.
10:25	<b>Start</b> thinking
10:28	min possible distance is $seq[0]-a_0$
10:28	max possible distance is $seq[0]+1$
10:30	E.g how many iterations are there between 6 and 7 while increasing by $1e-9$
10:31	try on eclipse
10:32	about $10^9$
10:34	increase by less than $10^9$ masalan

10:37	Binary Search :3
10:38	<b>Start</b> code sketching
10:46	<b>Start</b> coding
10:56	<b>Start</b> debugging
11:00	tracing a test case
11:05	Binary Search was a wrong idea
11:07	tried iterating by increasing by less than $1e-9$ , submitted, WA
11:09	tried another value, WA
11:11	think in other idea
11:14	editorial

### Log Analysis

I didn't verify 7etet el binary search f r7t f dahya+ I didn't brainstorm w 5adt awel fekra gatlly.

### SRM 434-2-2

Time	Logging
9:27	<b>Start</b> reading
9:30	<b>Start</b> thinking
9:30	some brute force solution I think
9:33	BF on all the lengths?
9:36	BF on all the perfect squares?
9:37	check if each perfect square exists and satisfies the conditions
9:37	This will be hard
9:37	try to form an arithmetic sequence from the table
9:38	max number will be 9 characters, then I can use an integer
9:43	<b>Start</b> code sketching

9:50	<b>Start</b> coding
9:53	<b>Start</b> debugging
10:03	done(arena disconnected and practise rooms are not available)

Log Analysis
I have to pay attention to all the values that my variables are going to be equal to while brute forcing.

## SRM 466-2-2

Time	Logging
1:56	<b>Start</b> reading
1:58	el7 string from 1 to 10 characters(can be converted to int)
2:00	<b>Start</b> thinking
2:03	BF?
2:07	I know how to get divisors in $O(\sqrt{n})$
2:14	factorize?
2:15	num of factors=num of divisors
2:16	no no that's wrong
2:18	trying to find a relation between num of divisors of x and num of divisors of $x*x$
2:22	The required number has to be of the same length of the string

2:26	don't know if it helps since it can have leading zeroes
2:27	no that doesn't help
2:31	check on pc winning tickets
2:34	They're all perfect squares
2:36	num of perfect squares isn't big
2:37	generate perfect squares in $O(\sqrt{n})$ and minimize
2:39	are there more perfect squares after $10^9$ till $10^{10}$ ?
2:40	$O(\sqrt{10^{10}})$ ?
2:43	<b>Start</b> coding
2:47	<b>Start</b> debugging
2:48	AC

**Log Analysis:** I should consider trying on PC later

## SRM 386-2-2

Time	Logging
6:27	<b>Start</b> reading
6:31	can't understand yet
6:38	<b>Start</b> thinking
6:41	nothing organised in my mind yet
6:44	generate all subsets?
6:46	only $2^{10}-1$ subsets
6:48	how to check if it's a superkey subset
6:51	ok now, how to check for the candidate superkey
6:52	I can imagine the code shwaya so no need for code sketch
6:53	<b>Start</b> coding

7:02	<b>Start</b> debugging
7:03	problems with the STL search function (compilation error)
7:13	now only one case that gives wrong answer
7:16	submitted, WA
7:19	I think my problem is misunderstanding some statements
7:30	pause for 3esha prayer
7:43	back debugging
7:48	editorial

Log Analysis
I used the STL search function for the 1st time + I didn't understand a statement correctly

## SRM 380-2-2

Time	Logging
4:09	<b>Start</b> reading
4:11	<b>Start</b> thinking
4:11	from constraints, I say it's a math problem
4:12	I have to think pictorially
4:31	simulating the moves to find a pattern or an equation
4:37	There's more than one equation due to different cases
4:41	<b>Start</b> coding
4:43	AC

## SRM 374-2-2

Time	Logging
6:28	<b>Start</b> reading
6:31	reading test cases
6:33	<b>Start</b> thinking
6:33	obviously it's a coding problem
6:34	I'll code sketch then
6:35	<b>Start</b> code sketching
6:39	<b>Start</b> coding
6:43	AC

## SRM 441-2-2

Time	Logging
10:43	<b>Start</b> reading
10:48	<b>Start</b> thinking
10:48	math problem I suppose
10:53	thinking pictorially
11:33	trying to organise the idea in form of equations
11:49	verify an equation on samples
11:51	There are more cases to handle
12:00	<b>Start</b> coding
12:04	<b>Start</b> debugging
12:07	AC

## SRM 433-2-2

Time	Logging
12:55	<b>Start</b> reading
12:59	<b>Start</b> thinking
1:00	normal simulation's order won't fit
1:05	trying to find some optimizations
1:09	thinking in some kind of precomputation
1:14	thinking in optimizing without using precomputation
1:33	I think I got a new way that is kind of easy
1:34	thinking about its order
1:36	thinking about how to implement this idea in the fastest way
1:43	<b>Start</b> coding



1:47	AC
------	----

Log Analysis
el fekra el sahla eli gebtaha fl a5er di kan el mafroud agebha badry 3an kda ya3ni

## SRM 483-2-2

Time	Logging
3:26	<b>Start</b> reading
3:31	how did he calculate test case 0?
3:35	<b>Start</b> thinking
3:35	I got the equation while in reading phase
3:36	I have to find a way to divide the equation into parts to avoid overflow
3:37	I think I got it
3:38	<b>Start</b> coding
3:42	WA
3:43	back to thinking
3:53	AC

Log Analysis
I had to check the corner cases before submitting! 7aga zay di maynfa3sh t7sal tani

## SRM 527-2-2

Time	Logging
11:42	<b>Start</b> reading
11:47	<b>Start</b> thinking
11:53	thinking pictorially
12:00	thinking in a greedy way to solve the problem
12:08	There's a sample that I don't understand
12:11	Ok got it but still no solution to the problem
12:15	fakes greedy, think BF
12:20	thinking recursively
12:29	dp
12:35	<b>Start</b> code sketching

12:39	<b>Start</b> coding
12:43	<b>Start</b> debugging
12:46	AC

Log Analysis
I dived gamed fl greedy w kda. I should've brainstormed.

## SRM 406-2-2

Time	Logging
1:49	<b>Start</b> reading
1:50	<b>Start</b> thinking
1:51	I need some observations
1:54	get all divisors with complexity $O(\sqrt{\max\_element\ in\ b})$ and check
1:56	<b>Start</b> coding
2:00	AC

Log Analysis
I could've taken any number not necessarily the maximum one. This little wrong assumption didn't affect me here but I should be more focused later.

I should've taken the min number to make the code faster.

## SRM 508-2-2

Time	Logging
4:08	<b>Start</b> reading
4:12	<b>Start</b> thinking
4:14	trying to get some observations
4:24	I know how to reach the solution but don't know yet how to get the minimum ans
4:28	verifying some idea on samples
4:33	my idea isn't complete yet
4:38	which p should I pick?
4:48	We go from 1 to N/p not from 1 to p
4:53	still don't know which number to pick

4:57	If I want to try them all, how can I do that?
5:02	editorial

## SRM 550-2-2

Time	Logging
9:00	<b>Start</b> reading
9:06	<b>Start</b> thinking
9:06	simulation problem
9:08	Imagining how can I get a dimension from the moves
9:10	beginning to think that there's some kind of greedy idea inside the simulation
9:13	I have a greedy approach that worked on some samples but I don't think it's 100% correct
9:15	Biggest grid's possible area would be $51 \times 51$
9:18	from which cell should I start?

9:19	Trying all cells as a starting point will fit
9:22	verifying and thinking about the code
9:32	Back to the old greedy idea, verify it
9:36	It works but needs improvement
9:38	apply the greedy idea then apply BF to check if ans is possible
9:39	<b>Start</b> code sketching
9:47	<b>Start</b> coding
9:56	<b>Start</b> debugging
10:06	stupid mistake, fixed it
10:06	still, there's an error in only one case
10:11	My code works as intended in that case
10:12	Back to thinking
10:14	I don't understand why the ans is -1 in that case
10:16	I got it
10:19	AC

Log Analysis	
I took a long time in debugging while the mistake was that I needed to swap n and m.	
I cycled when I understood the case in a wrong way. I understood why it outputs -1 when I got back to thinking and from the statement I understood why.	

## SRM 457-2-2

Time	Logging
8:33	<b>Start</b> reading
8:36	<b>Start</b> thinking
8:36	I'm thinking BF
8:41	minutes don't change with GMT change
8:44	I'm organizing my idea
8:51	<b>Start</b> code sketching
9:02	<b>Start</b> coding
9:08	<b>Start</b> debugging

9:21	AC
------	----

Log Analysis
My code could've been simpler

## SRM 387-2-2

Time	Logging
1:32	<b>Start</b> reading
1:36	<b>Start</b> thinking
1:36	It's a greedy problem I guess
1:41	observing the sample test cases
1:43	verifying a greedy idea on the samples
1:54	Brainstorming/verifying a lot of greedy ideas



2:06	Got an idea that works
2:10	<b>Start</b> coding
2:15	<b>Start</b> debugging
2:22	I fixed some stupid mistakes but I think that my idea doesn't handle the last case
2:23	Back to thinking
2:28	WA but passes on eclipse
2:30	No no, it also passes on topcoder. I thought that the case that gives WA is a sample. I submitted it again and gave WA again
2:32	editorial

Log Analysis
I had to be more organized while thinking. I had to think in reasonable/logical ideas, not just some random ideas and trying to verify them on samples.

## SRM 577-2-2

Time	Logging
10:08	<b>Start</b> reading
10:12	<b>Start</b> thinking
10:15	simulate it?
10:20	think about how to calculate the probability
10:23	I think I have a simple idea that works on samples
10:24	<b>Start</b> coding

10:31	<b>Start</b> debugging
10:32	Submitted
10:33	AC

Log Analysis
I asked myself while coding what if Elly's rating==MaxRating and I got back to the problem to check if there will be such input.

## SRM 539-2-2

Time	Logging
9:21	<b>Start</b> reading
9:24	<b>Start</b> thinking
9:24	I can generate all subsets using bitmasks
9:25	How to calculate the ans quickly while generating
9:27	How to handle the overlapping problem
9:30	I think I got it

9:31	<b>Start</b> coding
9:36	WA
9:41	<b>Start</b> debugging
9:45	WA
9:50	Bk to thinking
9:54	AC

Log Analysis
I didn't challenge myself enough

## SRM 486-2-2

Time	Logging
12:48	<b>Start</b> reading
12:51	<b>Start</b> thinking
12:51	dp?
12:54	I don't need to use the '-' operator at all
12:55	The '/' operator will always return 1
12:56	If I needed to use it, I have to use it only in the beginning

12:56	Apply recursion twice then. Once without '/' and once with it
12:57	<b>Start</b> coding
1:05	<b>Start</b> debugging
1:07	AC

## SRM 394-2-2

Time	Logging
3:58	<b>Start</b> reading
3:59	<b>Start</b> thinking
4:01	I have a dp idea
4:06	Do some brainstorming
4:11	I'm thinking about greedy ideas

4:38	I've got an idea. I'll verify it and then try to challenge it
4:44	<b>Start</b> coding
4:50	<b>Start</b> debugging
4:58	editorial

Log Analysis
I should try to consider normal BF(not recursive BF) because the solution wasn't greedy.

## SRM 382-2-2

Time	Logging
1:24	<b>Start</b> reading
1:27	<b>Start</b> thinking
1:27	The constraints are small
1:31	I'm thinking BFS
1:34	Brainstorm a little
1:40	I'm trying to solve the subproblem which is how to calculate minimum #of moves between a cell and another one

2:00	I don't have any proper ideas except for the BFS
2:03	I'm thinking how to implement BFS
2:16	editorial

Log Analysis
I'm not very good at BFS yet

## SRM 410-2-2

Time	Logging
5:15	<b>Start</b> reading
5:19	<b>Start</b> thinking
5:33	visit all the roads between each two cells connected to the main electrical grid
5:46	Put all the wires and then remove the ones that cause the disaster?
5:51	Subproblem: How to generate all the paths between two points?
5:57	verifying on a sample
6:00	How to implement such thing

6:14	I got an observation but I'm not sure if it's correct
6:17	<b>Start</b> code sketching
6:19	<b>Start</b> coding
6:23	WA,editorial

Log Analysis
I guess the samples were weak, I should've challenged myself

## SRM 453-2-2

Time	Logging
8:20	<b>Start</b> reading
8:25	<b>Start</b> thinking
8:26	The constraints are very small
8:30	I got a greedy idea
8:30	I'll verify it
8:32	no, forget it
8:47	Trying to think in a BF way

8:54	I think I got it
8:54	<b>Start</b> code sketching
8:57	<b>Start</b> coding
8:59	WA
9:21	editorial

Log Analysis
I've been told that this is a trivial problem so I was thinking BF/greedy. I thought about using bitmasks after getting WA and I didn't think about it good enough.

## SRM 500-2-2

Time	Logging
10:22	<b>Start</b> reading
10:35	<b>Start</b> thinking
10:42	simulate until I find overlapping?
10:58	thinking on sample test cases
11:12	trying to know how to simulate
11:20	I'm so stuck
11:24	editorial



## SRM 541-2-2

Time	Logging
1:35	<b>Start</b> reading
1:38	Tracing a test case
1:42	<b>Start</b> thinking
1:42	I was thinking while reading/tracing and I have an idea
1:43	Subproblem: how to find out if two points are going to meet. BF on all pairs and mark the ones that didn't find any other ant to meet
1:44	It will fit in time
1:44	verify on samples

1:46	I think marking/visiting the ones that have found their match will be easier
1:58	I'm beginning to doubt my idea
2:01	Brainstorm a little
2:02	I'll go ahead and try my idea, I don't think the other simulation by time idea will work
2:03	<b>Start</b> code sketching
2:09	<b>Start</b> coding
2:16	<b>Start</b> debugging
2:30	Problem in my idea, BK to thinking
2:33	BF on all the points and remove the ants that meet at it
2:34	coding
2:39	debugging
2:42	editorial

Log Analysis
I didn't take enough time in thinking/verifying and I dived.

## SRM 581-2-2

Time	Logging
12:47	<b>Start</b> reading
12:55	<b>Start</b> thinking
12:55	I'm thinking BF on all the segments but there're lots of conditions
1:01	I guess my problem is how to handle the overlapping of two cameras
1:03	Ok, I know how to solve the problem if there was no overlapping. Check the sample test cases where overlapping happens
1:11	I think my problem happens when there's more than one camera with same reports[i]

1:19	I think I got it
1:22	<b>Start</b> code sketching
1:35	<b>Start</b> coding
1:42	<b>Start</b> debugging
2:00	I had a problem in my idea
2:00	back to thinking
2:03	coding
2:08	debugging
2:22	back to thinking
2:40	weird behaviour in eclipse
2:41	I began to doubt that my code was working at some point but eclipse kan byhayyes
2:47	thinking
2:52	debugging
2:55	I passed all the samples now but I don't think this is the right solution
2:57	thinking
3:01	WA
3:19	transition

## SRM 400-2-2

Time	Logging
9:08	<b>Start</b> reading
9:10	<b>Start</b> thinking
9:16	I need some observations to optimize
9:18	a number where all the factors are the same and their number is $>1$ , it a strong prime power
9:29	I only need one check at the beginning if the $\text{sqrt}(n)$ is prime
9:30	what's my limit after that?

9:31	cubic root of $10^{18}$ =1million
9:31	sieve will fit
9:32	<b>Start</b> coding
9:39	WA
9:40	<b>Start</b> debugging
9:47	AC

Log Analysis
I should've revised my code to avoid that WA

## SRM 437-2-2

Time	Logging
11:57	<b>Start</b> reading
11:59	<b>Start</b> thinking
11:59	constraints are very small, next_perm will fit
12:04	The subProblem is how to transform a string to another in k swaps
12:05	what about making all the possible k swaps on my string?
12:05	I'll need recursion

12:07	<b>Start</b> coding
12:14	AC

## SRM 542-2-2

Time	Logging
4:31	<b>Start</b> reading
4:37	<b>Start</b> thinking
4:41	I need some observations
5:05	thinking symbolically
5:17	I'm stuck naw3an ma
5:19	editorial

## SRM 545-2-2

Time	Logging
6:39	<b>Start</b> reading
6:44	<b>Start</b> thinking
6:50	I'm thinking recursion
6:59	check if there's a greedy solution
7:09	How can I do this recursion?
7:19	thinking in a greedy solution again

7:28	verifying a greedy idea
7:33	<b>Start</b> coding
7:38	<b>Start</b> debugging
7:39	AC

## SRM 464-2-2

Time	Logging
10:20	<b>Start</b> reading
10:23	<b>Start</b> thinking
10:27	trying to get some observations
10:28	There are only 10 digits. If any of them occurred more than once. The string won't be colorful.
10:39	I'm trying to figure out how can I choose the numbers on which I'll apply next_permutation



10:40	generating them using bitmasks won't make them sorted
10:42	store them into a vector then sort?
10:45	another observation: the colorful string can't contain a zero
10:47	It also can't contain a 1
10:54	<b>Start</b> coding
11:02	AC

Log Analysis
I should be faster on getting the observations.

## SRM 467-2-2

Time	Logging
10:07	<b>Start</b> reading
10:16	<b>Start</b> thinking
10:25	trying to figure out how to calculate it
10:27	tb fl awal, get the intervals where the student is outside
10:31	I can actually simulate this problem

10:40	Trying to know how to simulate it
10:56	My solution won't work in case start==end. Think about how to handle it
11:03	It will return 1.0 or 0.0
11:08	<b>Start</b> code sketching
11:14	<b>Start</b> coding
11:19	<b>Start</b> debugging
11:21	AC

## SRM 561-2-2

Time	Logging
1:10	<b>Start</b> reading
1:18	<b>Start</b> thinking
1:23	The constraints for maxAccepted are small
1:26	how to use bitmasks to solve it?
1:31	shwaya medium w shwaya large

1:40	calculate with a function what happens in each case
1:41	<b>Start</b> coding
1:49	<b>Start</b> debugging
1:53	submitted
2:00	di tel3et WA o.O
2:01	debugging
2:07	AC

Log Analysis	
I stopped while coding to think about something	
i<mask(1<<n) while it should've been i<=mask(1<<n)	

## SRM 552-2-2

Time	Logging
2:27	<b>Start</b> reading
2:31	<b>Start</b> thinking
2:33	start thinking pictorially
2:36	At first, I need to get the minimum number of ingredients possible to paint a ballTriangle. And then, the rest is easy.

2:46	I know how to get num of balls in a ballTriangle
3:08	thinking pictorially
3:20	I don't know how to calculate
3:27	editorial

## SRM 531-2-2

Time	Logging
2:49	<b>Start</b> reading
2:50	I read this problem before
2:52	<b>Start</b> thinking
2:58	<b>Start</b> coding

3:02	<b>Start</b> debugging
3:04	AC

## SRM 360-2-2

Time	Logging
4:00	<b>Start</b> reading
4:05	<b>Start</b> thinking
4:07	I'm thinking about BF on all the possible numbers and check on the sum, but how?

4:14	try to get some observations from samples
4:20	There's some kind of common difference
4:21	Give your self some examples to check if that common diff still exists
4:23	I don't think it can be "CORRECT" without that common difference
4:24	<b>Start coding</b>
4:30	<b>Start</b> debugging
4:35	AC

Log Analysis
There are some observations in the editorial that I didn't observe

## SRM 540-2-2

Time	Logging
7:01	<b>Start</b> reading
7:06	<b>Start</b> thinking
7:08	dp?

7:11	try to construct the sequence?
7:11	numbers are from $-1e9$ to $1e9$ . need some optimization
7:21	I guess the sequence depends only on the 1st number in it
7:25	That 1st number is a positive integer. I can't loop on all positive integers
7:30	thinking symbolically
7:54	I reached something
7:55	pause (3esha)
8:11	bk
8:12	Forget about that thing that I've reached
8:22	editorial

## SRM 358-2-2

Time	Logging
9:35	<b>Start</b> reading
9:36	saw its solution before
9:37	<b>Start</b> coding

9:41	<b>Start</b> debugging
9:45	AC

## SRM 559-2-2

Time	Logging
11:12	<b>Start</b> reading
11:15	<b>Start</b> thinking
11:19	try to get some observations kda



11:30	7atta bl observations, maho ana msh ha cnt++ akid
11:37	thinking pictorially
11:45	I see some pattern
11:51	trying to figure out equations to calculate those patterns
12:00	<b>Start</b> coding
12:03	<b>Start</b> debugging
12:04	bk to thinking
12:05	I think the $k=6$ equation is wrong
12:07	fixed a case but still one case gives WA
12:23	got the $k=6$ equation
12:24	AC

Log Analysis
kan el mafrood a-make sure en all my equations are correct

## SRM 423-2-2

Time	Logging
12:52	<b>Start</b> reading
12:56	tracing some samples
1:00	<b>Start</b> thinking

1:01	is it ok to use an existing cell to store checkers all the time?
1:02	I think so
1:05	<b>Start</b> coding
1:11	<b>Start</b> debugging
1:13	a cell cannot exist and I can use it :/
1:13	bk to thinking
1:27	I can try to loop on all cells not just the ones with the same index
1:31	coding
1:33	AC

Log Analysis	
I started solving using a wrong idea. I should've verified it better.	

## SRM 488-2-2

Time	Logging
6:45	<b>Start</b> reading
6:52	<b>Start</b> thinking
6:54	can I just calculate all the substrings from the 2 strings and then work on them

6:55	How will I work on them?
6:56	I guess the order will fit
6:57	organize the idea shwaya kda tb
7:02	<b>Start</b> coding
7:06	<b>Start</b> debugging (weird error)
7:08	It was just a segmentation fault. I wrote i instead of j. but still there's one sample with WA
7:14	bk to thinking. There's a problem in my idea
7:23	coding
7:28	debugging
7:32	bk to thinking
7:32	pause
10:55	back(thinking)
10:58	tb how can I use bitmasks masalan to solve this one?
11:00	tb fakes bitmasks delwa2ty
11:06	coding
11:15	debugging
11:16	time limit on a sample, thinking
11:22	coding
11:25	thinking
11:29	editorial

## SRM 594-2-2

Time	Logging
1:56	<b>Start</b> reading
2:01	<b>Start</b> thinking
2:05	3ayz a2af 3ala kol pair of ratio f A wa-check all the other pairs f B

2:05	check if this idea is true
2:22	I'm trying to figure out how to calculate the ans by hand
2:31	I remember getting stuck on how the ans is being calculated while I was in that SRM
2:32	editorial

## SRM 448-2-2

Time	Logging
3:22	<b>Start</b> reading
3:25	<b>Start</b> thinking
3:25	get some observations

3:29	After some shuffling the sequence repeats itself.
3:32	when does the sequence repeat itself?
3:36	I'll try to simulate until I find the place where it repeats itself and stop
3:39	Pause
4:09	back
4:09	check larger examples kda
4:15	badawar 3ala pattern
4:20	If I want to use modulus, how will I use it?
4:24	I don't know when the sequence will repeat itself, try to simulate it on PC
4:32	I don't think I can use the fact that the sequence repeats itself in that way
4:39	trying to find a pattern
4:46	can't find one. think about how to get the top card only within each shuffle
4:59	<b>Start</b> coding
5:01	AC

Log Analysis
Instead of wasting time on tracing something on paper. Just simulate it on PC.

## SRM 449-2-2

Time	Logging
6:10	<b>Start</b> reading
6:11	<b>Start</b> thinking

6:11	I remember its solution
6:11	<b>Start</b> coding
6:13	AC

## SRM 517-2-2

Time	Logging
8:10	<b>Start</b> reading
8:12	<b>Start</b> thinking

8:12	check first if it's not prime and then get all the pairs
8:14	decompose all the pairs
8:22	<b>Start</b> coding
8:29	<b>Start</b> debugging
8:33	WA
8:33	debugging
8:38	bk to thinking. I don't understand why sample test 7 outputs no
8:39	Ohhh
8:40	aw la2a keep thinking why
8:42	I got it.
8:46	coding
8:50	debugging
9:20	editorial

Log Analysis	
I rushed into coding a lot. I should've understood the problem correctly.	

## SRM 580-2-2

Time	Logging
12:34	<b>Start</b> reading
12:37	<b>Start</b> thinking



12:38	I remember this one
12:40	<b>Start</b> coding
12:41	AC

## SRM 583-2-2

Time	Logging
12:58	<b>Start</b> reading
1:04	<b>Start</b> thinking

1:08	It's a coding problem
1:09	<b>Start</b> code sketching
1:14	pause
1:16	bk
1:17	<b>Start</b> coding
1:32	<b>Start</b> debugging
1:34	AC

Log Analysis
fih 7etta ma2arethash kwayes f l started debugging, law a5er sample makanetsh mawgooda knt hatsawa7 b WA maloosh lazma.

## SRM 511-2-2

Time	Logging
1:59	<b>Start</b> reading

2:01	<b>Start</b> thinking
2:09	<b>Start</b> coding
2:12	AC

Log Analysis	
asl I solved it before	

## SRM 512-2-2

Time	Logging
10:37	<b>Start</b> reading

10:40	<b>Start</b> thinking
10:42	BF on all the possible answers and check on each one if valid, then get the max
10:45	<b>Start</b> coding
10:51	<b>Start</b> debugging
10:54	AC

## SRM 525-2-2

Time	Logging
9:59	<b>Start</b> reading

10:00	<b>Start</b> thinking
10:04	BF on all the rectangles
10:07	<b>Start</b> coding
10:14	AC

## SRM 510-2-2

Time	Logging
10:45	<b>Start</b> reading

10:48	<b>Start</b> thinking
10:48	BF on everything and maximize the answer, don't minimize
10:50	<b>Start</b> coding
10:54	<b>Start</b> debugging
10:58	WA
10:59	bk to thinking
11:04	coding
11:06	debugging
11:10	AC

Log Analysis
I didn't calculate the order

## SRM 557-2-2

Time	Logging
4:30	<b>Start</b> reading

4:35	<b>Start</b> thinking
4:40	try all the scenarios masalan?
4:42	I was thinking about bitmasks bs I think a scenario could be bl 3aks
4:47	I think I can go ahead and use it
4:50	<b>Start</b> code sketching
4:54	<b>Start</b> coding
4:58	WA
4:58	thinking
4:58	I have another idea kda
5:00	coding
5:02	AC

## SRM 602-2-2

Time	Logging
------	---------

9:27	<b>Start</b> reading
9:29	<b>Start</b> thinking
9:29	I remember this one :3
9:31	<b>Start</b> coding
9:33	<b>Start</b> debugging
9:35	AC

## SRM 543-2-2

Time	Logging
------	---------



12:52	<b>Start</b> reading
12:53	<b>Start</b> thinking
12:55	nel3ab fl binary representation bta3t el numbers?
12:57	try on PC manzar their binary representations kda
1:04	I need observations
1:10	try to get from PC some properties of xor
1:17	xoring intervals of 4 numbers=0
1:20	thinking how to use this observation
1:25	<b>Start</b> coding
1:27	<b>Start</b> debugging
1:28	AC

## SRM 588-2-2

Time	Logging
------	---------

10:46	<b>Start</b> reading
10:49	<b>Start</b> thinking
10:49	I'm thinking bitmasks
10:50	<b>Start</b> coding
10:53	bk to thinking, doesn't pass last test case
10:54	sorting the vector inside each loop of bitmask doesn't make it pass bardo
10:57	tb try kda sorting fl awal
10:57	coding
10:59	AC

Log Analysis	
I was trying to sort marra 3al pair el awal w marra 3al tany. el tany worked f 3amalt submit. Try to be more logical ya3ni in choosing a solution.	
I already doubted fl awal en I should sort according to el tani	

## SRM 409-2-2

Time	Logging
------	---------

11:26	<b>Start</b> reading
11:30	<b>Start</b> thinking
11:32	observing
11:36	developing an idea
11:42	pause
11:53	bk
11:54	thinking why's the answer in last sample is 12
11:55	got it
11:55	verifying my idea on samples
12:00	adding some stuff to the idea
12:02	<b>Start</b> coding
12:11	<b>Start</b> debugging
12:13	WA
12:13	pause
12:29	bk (thinking)
12:30	I think using string.find cause the WA
12:30	think in an alternative
12:37	coding
12:44	debugging
12:50	editorial

Log Analysis	
I rushed a lot	

## SRM 513-2-2

Time	Logging
------	---------

2:08	<b>Start</b> reading
2:18	<b>Start</b> thinking
2:22	If there's a ball that has the same coordinate return 0
2:33	try to verify an idea kda
2:36	<b>Start</b> coding
2:39	<b>Start</b> debugging
2:42	AC

## SRM 589-2-2

Time	Logging
------	---------

8:30	<b>Start</b> reading
8:31	<b>Start</b> thinking
8:31	I remember this one
8:36	<b>Start</b> coding
8:38	<b>Start</b> debugging
8:43	AC

## SRM 584-2-2

Time	Logging
------	---------

9:07	<b>Start</b> reading
9:09	<b>Start</b> thinking
9:12	I'm thinking graph
9:22	I don't know how to implement it
9:30	el -1 da hayrga3 f 7alet en el graph msh connected bs
9:34	7ot el friends f adjaceny list lateefa kda w3mel el BFS el lazeez
9:45	editorial

## SRM 407-2-2

Time	Logging
------	---------

12:45	<b>Start</b> reading
12:48	<b>Start</b> thinking
12:54	I return long long which is confusing shwaya
12:57	adawar 3ali ma3andoosh subordinates el awal
12:59	n3mel adjacency list w nshta3'al 3ali 3ando as3'ar 3adad nodes el awal msalan?
1:04	<b>Start</b> coding
1:09	there's something I didn't think about,bk to thinking
1:13	coding
1:17	<b>Start</b> debugging
1:24	only one test that's not correct, bk to thinking
1:27	try an idea kda
1:28	coding
1:34	WA,editorial

Log Analysis	
I doubted my idea when I got WA, the problem was just that I didn't return long long	
I'll add a new thing to my discipling called "code brooming" :D	

## SRM 579-2-2

Time	Logging
------	---------

10:48	<b>Start</b> reading
10:56	I can't understand something in a test case
11:04	<b>Start</b> thinking
11:10	verifying an idea kda
11:17	<b>Start</b> code sketching
11:29	<b>Start</b> coding
11:39	<b>Start</b> debugging
11:39	only last sample that doesn't pass
11:52	my logic doesn't pass the last sample, bk to thinking
11:55	debugging
11:59	AC

Log Analysis	
I wasted a lot of time	

## SRM 551-2-2

Time	Logging
------	---------



12:56	<b>Start</b> reading
12:58	<b>Start</b> thinking
1:02	I have a BF+greedy idea kda bs not sure lessa
1:04	trying to challenge myself
1:09	<b>Start</b> coding
1:13	<b>Start</b> debugging, only one sample that doesn't pass
1:15	I understood something 3'alat
1:16	bk to thinking
1:20	coding
1:21	no no bk to thinking a7san
1:30	coding
1:40	debugging
1:42	WA,editorial

Log Analysis
more focus on reading the problem statement

## SRM 599-2-2

Time	Logging
------	---------

3:10	<b>Start</b> reading
3:10	<b>Start</b> thinking
3:10	factorize, I remember this one
3:11	use map for the power
3:12	<b>Start</b> coding
3:15	<b>Start</b> debugging
3:21	AC

## SRM 590-2-2

Time	Logging
------	---------

4:54	<b>Start</b> reading
4:57	<b>Start</b> thinking
4:59	I'll do BF but thinking how
5:04	ageeb all connected components first wa3allem 3ala kol wa7ed b num fl vis
5:05	agarrab kol cell fadya a7ot fiha black dot wamshy 3ala kol connected component
5:06	<b>Start</b> code sketching
5:15	<b>Start</b> coding
5:25	<b>Start</b> debugging
5:39	bk to thinking
5:40	coding
5:44	debugging
5:50	AC

Log Analysis
fih 7aga ma5adtesh baly menha

## SRM 566-2-2

Time	Logging
------	---------

6:09	<b>Start</b> reading
6:11	<b>Start</b> thinking
6:16	bada2t afham shwaya ezay ha-approach it
6:21	el fekra bs en mafish 7aga t-overlap
6:24	max number of pairs is $n-1$
6:27	bfakkar a7otohom f vector of pairs wa3melohom sort 3ala 7asab el length
6:30	pause
6:55	bk
7:02	ezay ageeb max num of pairs
7:08	fakkar greedy shwaya
7:16	tb nwasal kol wa7ed l a2rab 7aga leeh w b3d kda nsheel eli 3amal overlap
7:18	<b>Start</b> coding
7:27	<b>Start</b> debugging
7:29	editorial

Log Analysis
I had to think more logically in greedy

## SRM 535-2-2

Time	Logging
------	---------

9:47	<b>Start</b> reading
9:48	<b>Start</b> thinking
9:50	think symbolically
9:54	<b>Start</b> coding
9:56	AC

## SRM 493-2-2

Time	Logging
------	---------

10:28	<b>Start</b> reading
10:31	<b>Start</b> thinking
10:40	I can't even implement its dp :D
11:01	editorial -_-

## SRM 558-2-2

Time	Logging
------	---------

11:27	<b>Start</b> reading
11:33	<b>Start</b> thinking
11:41	try all possible lengths and then recurse on all possible positions
11:52	I need conditions in my dp
12:01	<b>Start</b> code sketching
12:09	<b>Start</b> coding
12:17	<b>Start</b> debugging
12:29	editorial

## SRM 462-2-2

Time	Logging
------	---------

1:00	<b>Start</b> reading
1:04	<b>Start</b> thinking
1:04	BF on all the bases
1:07	BS
1:09	think about -1 and -2 cases
1:13	<b>Start</b> coding
1:19	testing my code for the -1 and the -2 cases (thinking)
1:22	There's a case that returns 0 -_-
1:23	add the condition
1:24	AC

## SRM 356-2-2

Time	Logging
------	---------



1:32	<b>Start</b> reading
1:35	<b>Start</b> thinking
1:45	Problem simplification: e3tebro int
1:55	I'm thinking of trying all the values between 0 and 10 using bitmasks
1:57	check if I can transform a double into a string using stringstream
2:03	<b>Start</b> coding
2:09	<b>Start</b> debugging
2:21	I was working 3'alat aslan fih 7aga nsetha bayen aw markztesh fiha :S
2:22	editorial

## SRM 544-2-2

Time	Logging
------	---------

4:51	<b>Start</b> reading
4:54	<b>Start</b> thinking
4:55	think greedy
5:00	think about simulating it
5:03	I'll go ahead and try the idea
5:04	<b>Start</b> coding
5:06	AC