

# IN CONVERSATION WITH

# TESLA

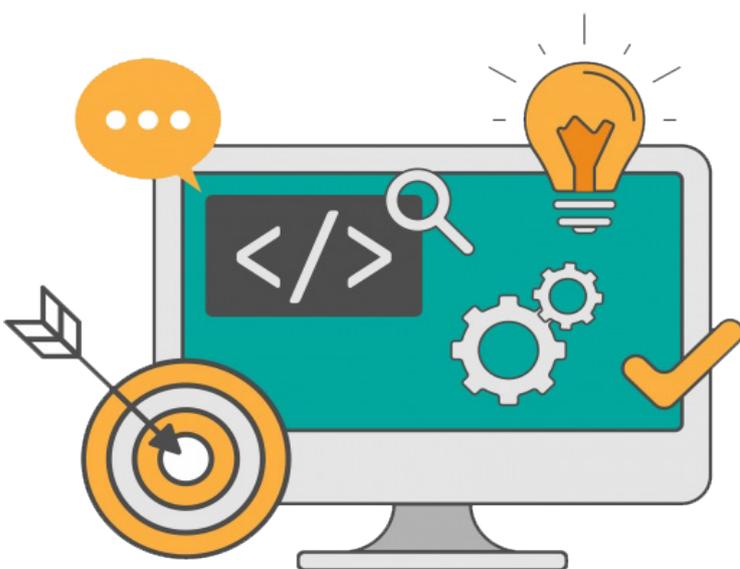
# PROTOCOL



- Ujwal Narayan

ICPC is one of the things that nobody in IIIT can escape hearing about. With computer programming and data structures being present right in the first year, everyone knows about the skills needed for a team to go to that level. IIIT has had an amazing history in its short time, with it having sent teams almost every year since 2012. This year too we are proud to announce that, IIIT has a team, “tesla\_protocol” comprising Anurudh Peduri, UG5, Arjun P, UG 5 and Devansh Gautam, UG4 who will all be travelling to Moscow to represent India on the biggest platform there is.

I had the pleasure of talking with Anurudh who was kind enough to spare a few moments to talk about his experience with ICPC this year. Given below is a few excerpts from our discussions.



## How was ICPC this year for you?

We were performing decently in the practice contests. Our aim was to come 1st so with respect to that we were not doing that great. Among the top teams, we usually have the most number of wrong submissions and hence a large number of penalties. Keeping all this in mind we knew we had to solve what everybody else solved plus an additional one to make up for our wrong submissions. We needed not just ability but also the speed to solve the hard problems before time ran out.

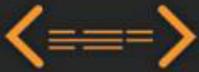
Our first regionals was at Amrita, and we messed up. It was the first contest we gave with all three of us in the same room. (Arjun during this time was interning at ETH, Zurich) Perhaps because of this, we were not as coordinated as we could have been. There were communication gaps, and we made some strategic errors especially when deciding which problem to solve. Even though the problems were easy, due to the contest pressure we were a little confused. Do we tackle problems as a team or do we let individual members play to their strengths and solve whatever problems they can? There are pros and cons to both and we did not get the right approach within that time frame. We finally ended up 6th.

Kanpur was our second regionals. Wrong submissions haunted us again. At some point we just started panic submitting, coming to use the Online Judge as a debugging tool. Each submission identified a new mistake for us to fix. And all these penalties started to add up. Other than this, we were also slightly out of touch for a few concepts and that made it difficult to quickly solve some medium level problems.

All this paints a bleak picture, but the only thing we could do was practice. Arjun and I were both in our 5th year, and it was our last attempt at ICPC. It was a do or die situation. We reevaluated our approach. We took a good hard look at the

# COMPETITIVE PROGRAMMING

//Where Computer Geeks become Gladiators



mistakes we made and focussed on correcting those errors. I'd say we were much more prepared especially mentally for this.

But as they say, no plan survives first contact with the enemy and so was the case here. The contest was very ad-hoc. It boiled down to a speed contest and penalties made or broke teams. In the first 1.5 hours of the contest, we could only solve 3 problems. To give context, the leading team at this point had solved 6 problems. I was looking at the list of problems totally dejected. It was a do or die situation for us and we were almost dead. (If this was an anime, cue melancholy music) Devansh, however, picked a problem with no solves yet and said it was easy. And before too long he had solved it. This gave us motivation and we solved another easy problem. Devansh again picked up an unsolved problem (This later turned out to be the hardest solved problem in the contest with only 4 teams solving it). While it was not as easy to slay as the earlier one, he coded and debugged it quickly and had solved it as well. Before the rank list froze (In contests such as these, the leaderboard does not change in the last hour and thus teams cannot see how many problems the other teams have solved during that time or their final rank) we had solved 7 problems, ranking 10th. We were outside the qualification zone for sure at this juncture but the motivation was sky high. Since we had solved the hard problems first, we had relatively easy ones left

so we knew that if we focus, we'd solve them and qualify. We solved one more 20 minutes after the freeze. With around 3 minutes remaining before the contest ended, we submitted our last solution, and thus solved 9 problems in total. Due to all our penalties, we knew we'd come last amongst the teams that solved 9 problems, so even if the cut off was at 9 problems we might come out on the wrong end of the qualification zone. The centre had poor internet connection and thus it wasn't possible for us to figure out how the other teams did easily. It was still not a done deal and thus we were on the edge. The rank list finally came out and we were 7th. Only the top 6 can qualify. But three of the teams who were ranked above us had already qualified for the World Finals earlier through the regionals. In the end, it seemed the team that solved 8 problems with the lowest penalty qualified as well, so we "comfortably" made it through.

So yeah, we qualified, so it was good I guess!

**How did tesla\_protocol prepare for ICPC? Was it very different from last year? Any particular insights?**

I think we were more serious about ICPC this year when compared to our earlier attempts. Like last year (tesla\_protocol

consisted of Anurudh, Arjun and Rishabh Arora), for example, we did not give enough team contests together and it showed. We found it difficult to coordinate and work together during the contests. This year even though we may be separated by thousands of kilometres (Arjun was in ETH, Zurich at the time) we made sure to give team contests regularly. Being a 5th year, I was not as burdened by the academic load as before and could devote more time to the things that I wanted to do. While we were serious, we were not as serious as we should be. But by Mid September, however, we realized we actually had a pretty good shot at making it to the world finals.

I believe it's essential to have a good rapport with your teammates. ICPC is after all a team contest. Communication becomes key here. It's important to know your strengths and weaknesses but also your teammates' strengths and weaknesses so you can accurately judge who is best suited to solve a problem. In ICPC especially there is just one computer, so each minute you spend on it becomes precious. You also will have large gaps between your usage as your teammates will also need to use it to code. So you can't always immediately make changes and debug. Details also may slip your mind and not even we are exempt. For example, in the last problem for West Asia, we messed up with the array indexing. I had written half the code at the beginning of the contest, and at the end when we came back to it we thought the indexing was another way. I found the bug right after the contest ended, but alas, it was too late.

Working simultaneously is important here. Trust also plays a role as while you trust your teammates to solve a problem, you must also trust them to be open about the difficulties they are facing if they cannot solve it. Being able to come forward and explain why I'm stuck on a problem without fearing judgement from my teammates is crucial.

We were also lucky with each of us being able to cover each other's weaknesses. Most topics had a good overlap with the members who knew it well and there were no gaps in the team. While a certain



minimum skill is needed and expected at this level, topic coverage and team dynamics is also important to consider when making a team. While finding such a set of teammates is hard, it's not always necessary as solving all the problems is not necessary to qualify for ICPC. Teams can work around this if they have a good dynamic.

**You've emphasised team practice quite a bit. How different is it from individual practices?**



With a team, you can solve much harder problems. The whole is greater than the sum of its parts.

We can discuss with each other, and bounce ideas off of each other. Discussion brings new perspectives and it becomes easier to find mistakes and debug. Questions that need expertise in different topics can be split into individual topics where each member solves the topic he's good at. So while individually we could not have solved the complex problem, as a team we could.

Other than this, the environment is different and team members can offer emotional support and encouragement as well. Teammates are able to motivate each other and help each other to a certain extent. Team spirit is underrated. It is possible that a team where all of the team members are stronger individually than most teams can not make it to world finals because of the lack of synergy. The stakes are high, and tempers rise. It's not uncommon for arguments to happen which of course not just wastes time but also distracts.

All said and done, it essentially boils down to whether the teammates can work with each other and understand each other. If yes, then with enough practice you should be as well equipped in this area as anybody else.

**Is being in IIIT advantageous? Or do the academic requirements prove to be a burden?**

IIIT has a very good programming culture. ICPC has a lot of interest and we have plenty of good role models to look up to.

The Programming Club deserves a lot of credit for making sure that the culture is not just carried forward but actively growing. A lot of meets are conducted with a variety of topics at a variety

of levels right from beginner to advanced being taught and discussed. The peer-based learning approach we use here, with the students themselves teaching each other and helping each other helps to develop your skills to greater heights. These days, the Programming Club also has challenges for regular contests between members which help to keep one's skills sharp and keep you in touch with competitive programming even if you are not actively pursuing it.

Monsoon semester usually faces a decline in practice as it is right after the summer which essentially breaks the flow. With deadlines every week, a team practice of 5 hours is a commitment that is hard to do week in week out. The college faculty are also not super supportive and it was the peer-based



learning that helped us develop our skills beyond the basic data structures.

When you have peers and friends to discuss and solve with, you don't feel as if you are doing something that is work. Friends can also motivate you to be better than what you are. Major props and thanks to the Programming Club and its coordinators for working so hard with little to no benefits just so that the culture continues.

**There are a lot of platforms like Codeforces and Codechef out there. Which one do you prefer to practice with? Are there any major differences other than the UI between those sites?**

The problem styles and topic coverage vary across different platforms. AtCoder is usually math-oriented, and has problems with more elegant solutions, Codeforces strikes a balance between implementation level and the topics covered. TopCoder has high complexity solutions and has a wide

variety of combinatorics problems. Codechef has a variety of hard and unusual problems. So all the different platforms have their own distinct styles and make you think in different ways to get the answer. ICPC, on the other hand, gives you all of these in one contest. Codeforces rating is somewhat misleading as you can achieve red with a narrow breadth of topics.



What is important is not the platform, but the quality of problems and so it's important to do good problems irrespective of the platform. We usually gave team contests in the Codeforces gym and practised with the various ICPC regionals from around the world as well. At the later stages we also practised with OpenCup which had problems which were approximately of a similar difficulty to what we would expect in the world finals.

**The leaderboard is a staple of competitive programming. Does it provide any tangible benefits other than knowing who's on top? Teams are liable to become nervous or conversely too confident by looking at their positions on the leaderboard, so won't it be better if they do not look at it at all?**

I think the leaderboard is extremely useful. It helps you identify what problems everybody is solving and can serve as an approximation of the difficulty. Not just this, but by looking at the leaderboard you can see if a particular problem is implementation heavy or if it has a clever trick which you can exploit. For example, if a problem is solved by teams across the leaderboard, more often than not there's a pattern which has to be observed and used to solve it. So it's probably better for you to move on to another problem rather than placing a bet on a problem whose solution may or may not strike you.

### On competitive coding and beyond...

Competitive Programming has many benefits. You acquire a solid grasp on your language of choice and will be intimately familiar with all its functionalities and intricacies. As the



solutions need to be fast and efficient to be judged right, you will also be pretty good with respect to algorithms and data structures.

But this is just the tip of the iceberg. Competitive coding makes you think and apply your skills to solve a novel problem. So it doesn't just train you in coding, but also in the art of problem-solving. It's a mental exercise and you need to use whatever you know to come up with a solution, and as there's a time limit you need to be able to do this fast.

I think of it as a sport. Each problem solved to me is roughly equivalent to a goal scored by a striker. I do competitor programming because I like it and I think that's an important distinction to make. Many do CP due to the advantages it offers be it during internships or jobs, and it's perfectly fine to do so. Many of them, in fact, start out because of the above reasons but end up continuing because they like CP and not due to

any external factors. One shouldn't feel forced to do CP just because everyone does it or to make more money. Give it a try outside of OJ a couple of times and if it's not your cup of tea don't feel pressured to continue. As I said, it's just like any other sport. Just because you're a poor competing coder does not mean you're a poor coder or a researcher. Everybody has their own interests and this just happens to be mine.

So what I'd like to say is don't let OJ be your only experience with regards to CP. Try it a couple of times outside of academic pressure and who knows you may just grow to like it!

On behalf of all of us here at IIIT, we'd like to extend our congratulations and would like to wish `tesla_protocol` the best in all their future endeavours including but not limited to the World finals! ■

