

# ACM-ICPC Presentation

---

Lionel KITIHOUN

Part of the content comes from some slides found on the web. I do not know the authors, so here are the links to the slides.

- <https://slideplayer.com/slide/1614833>.
- <https://slideplayer.com/slide/1615102>.
- <https://slideplayer.com/slide/8333198>.

# Summary

Presentation of the contest

Preparation

Registration

Training

Questions

## **Presentation of the contest**

---

- ACM International Collegiate Programming Contest is probably the greatest university-level programming contest in the world.
- ACM - Association for Computing Machinery.
  - World's largest scientific and educational computing society.
  - Organizes research conferences.
  - Publishes journal.
- The contest is organized since 1977.



*Credit: Wikipedia*



*Credit: Wikipedia*

# Purpose

- Provide students the opportunity to sharpen problem solving skills.
- It is not just about programming.



Multi-tiered contest.

- Local contest, where each university selects its best teams.
- Regional contest (semi-finals).
- World finals.

- Africa and Arab region.
- The regional contest direction is in Egypt.
- Regional site for West Africa is in Benin.
- Contest direction in Benin, Togo, Niger, and Burkina-Faso is under by Mapcom Group responsibility.

- Each year, a university is given the honor to host the finals.
- It is a great privilege.
- Reaching the finals is already an accomplishment.

# Rules

- Each team consists of three (3) students from the same institute or university.
- Each team has access to one computer during the contest.
- The contest lasts five (5) hours and there are eight (8) or more problems to solve.

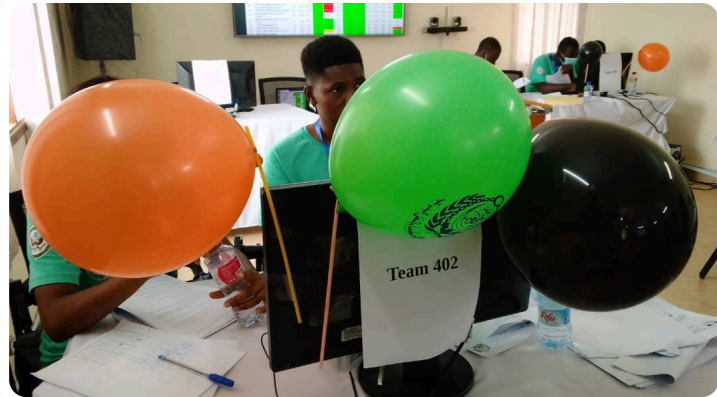
## Restrictions and quotas

- Each student should have less than five (5) years of university education before the contest.
- Students who have previously competed in five (5) regional contests or two (2) world finals are not illegible.
- A university can have several teams qualified to a regional contest, but there is at most one team per university at the world finals.
- Read the rules at <https://icpc.global/regionals/rules>.
- Allowed languages are C, C++, Java, and Python.

## Contest process

- There can be several sites for a regional contest, but there is only one for the world finals.
- On each site, teams are gathered in a room.
- There is a software that manages the contest.
  - It allows you to submit your solutions.
  - You get back the results of your submissions.
  - You can ask questions to the judges if you think something is unclear about a problem.
- There is a scoreboard.
- Each time a team solves a problem, it gets a balloon.

## A team with its balloons



*Credit: Facebook*

## A contest room



*Credit: news.itmo.ru*



# World Finals - Porto 2019



*Credit: [icpc2019.up.pt](http://icpc2019.up.pt)*

- First, teams are ranked based on the number of solved problems.
- When two or more teams have the same number of solved problems, they are further ranked by time penalty.
  - For each solved problem, the number of minutes from the beginning of the contest until the correct solution was submitted.
  - For each solved problem, twenty (20) minutes for each incorrect submission before the correct solution.

## Possible answers to submissions

- **Accepted.**
- **Wrong Answer (WA).** Your program does not give the correct answer.
- **Time Limit Exceeded (TLE).** Your program takes too much time. That does not mean it gives the correct answer.
- **Memory Limit Exceeded.** Your program uses too much memory. Same remark as above.
- **Runtime Error.**
- **Compilation Error.** No time penalty for this type of errors.

Here are some tips.

- Good teamwork is essential.
  - You must be complementary and know how to work together.
  - Each member should have one or several specialties.
  - Ideally, everyone should have a good programming level.
  - There is only one computer available, so it must be shared.
- Once you think you have a solution to a problem, don't rush to the computer. Write solution outline on a paper. It helps a lot.
- Identify easy problems and solve them first.
- Learn to relax and don't explode under pressure.
- Enjoy the moment.

# Problem categories

All areas of mathematics.

- Geometry.
- Graph Theory.
- String processing.
- Calculus.
- Optimization.
- etc.

# Preparation

---

**Preparation**

---

**Registration**

# Registration

In order to compete, students and coach should be registered on ICPC website.

Once your account is created, you must complete your profile.

- Name, birthdate, sex, etc.
- Academic information.
  - Your university or institute.
  - Post-secondary studies start date.
  - Branch.
  - Expected date of graduation.



With your account, you can register for a contest.

- One of the team member or the coach creates the team and add members.
- The contest organizers validate each team.
- If any of the team member did not complete its profile or does not fulfill the requirements, the team will be rejected.

# Preparation

---

## Training

## An endless, beautiful journey

- You can't be an ace right from the start.
- You will need a lot of training to reach a decent level.
- It never ends. There will be always something to learn.
- It is not about winning the contest, it's all about improving and becoming a good problem solver.
- I am not saying that it is not cool to win the contest.

## The most important thing

You must have a strong will to learn and get better. It will fuel you through your journey.

*Success Is Going from Failure to Failure Without Losing Your Enthusiasm.*

—William Chruchill

## Basic prerequisites

- English.
- You must know how to use a computer.
- The computers used during the contest run under Linux, so it's better to get used to it.
- You must know at least one of the languages of the contest.
- You must know how to compile and run programs from the command-line.
- You must also know how I/O redirection works.

## More prerequisites

- DSA course.
  - You must know common data structures and algorithms.
  - You must be familiar with the notion of algorithm complexity.
- Good level in C++, Java, or Python.
  - Object-oriented programming (optional, but useful).
  - You must know your preferred language strengths and weaknesses.
  - And its standard library.
- Fast typewriting is an advantage.

# Learning path

- Follow a competitive programming course.
  - Usually, universities have a dedicated course and competitive programming club.
  - Some websites offer training material (cp-algorithms.com, Geeks for Geeks).
- Read books and tutorial on recurrent topics. There is a lot of material available on the web.
- Practice and learn on competitive programming websites.
  - Codeforces.
  - Kattis.
  - Hackerrank.
- Participate in contests.
  - ACM Contests.
  - Google Codejam, Kickstart, Hashcode.
  - Facebook Hacker Cup.
- Practice and learn, over and over again.

# How will we work?

- We will use Kattis (<https://open.kattis.com>).
  - Create an account if you don't have one.
  - Read the documentation for your language in the help section.
- T-414-ÁFLV: A Competitive Programming Course, by Bjarki Ágúst Guðmundsson. Available at <https://algo.is>.
- We have a WhatsApp group and a Slack workspace.



# Course topics

- Data structures.
- Problem solving paradigms.
- Greedy algorithms.
- Dynamic programming.
- Graphs.
- Mathematics.
- Strings.
- Geometry.

## Problem set

For each chapter, there are five problems to solve and two bonus problems.

## Questions

---

