Problem M. Preparing a contest

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	2048 mebibytes

The 57th **F**lower **O**lympiad in **I**nformatics (LVII **FOI**) will be organized by **C**entral **F**lower **C**omputing **S**ectors in 2022.

As the chief judge of **FOI**, Little Q will have the opportunity to take full responsibility for the development of the FOI contest problems. Although she doesn't have experience running competitions, don't worry - she has a full five hours to complete these preparations.

What is Competitive Programming?

Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications. Contestants are referred to as sport programmers. Competitive programming is recognized and supported by several multinational software and Internet companies, such as Google and Facebook.

A programming competition generally involves the host presenting a set of logical or mathematical problems, also known as puzzles, to the contestants (who can vary in number from tens to several thousands), and contestants are required to write computer programs capable of solving each problem. Judging is based mostly upon number of problems solved and time spent for writing successful solutions, but may also include other factors (quality of output produced, execution time, program size, etc.)

One of the oldest contests known is ICPC which originated in the 1970s, and has grown to include 88 countries in its 2011 edition.

From 1990 to 1994, Owen Astrachan, Vivek Khera and David Kotz ran one of the first distributed, internet-based programming contests inspired by ICPC.

Interest in competitive programming has grown extensively since 2000, and is strongly connected to the growth of the Internet, which facilitates holding international contests online, eliminating geographical problems.

The aim of competitive programming is to write source code of computer programs which are able to solve given problems. A vast majority of problems appearing in programming contests are mathematical or logical in nature. Typical such tasks belong to one of the following categories: combinatorics, number theory, graph theory, algorithmic game theory, computational geometry, string analysis and data structures. Problems related to constraint programming and artificial intelligence are also popular in certain competitions.

Irrespective of the problem category, the process of solving a problem can be divided into two broad steps: constructing an efficient algorithm, and implementing the algorithm in a suitable programming language (the set of programming languages allowed varies from contest to contest). These are the two most commonly tested skills in programming competitions.

In most contests, the judging is done automatically by host machines, commonly known as judges. Every solution submitted by a contestant is run on the judge against a set of (usually secret) test cases. Normally, contest problems have an all-or-none marking system, meaning that a solution is "Accepted"only if it produces satisfactory results on all test cases run by the judge, and rejected otherwise. However, some contest problems may allow for partial scoring, depending on the number of test cases passed, the quality of the results, or some other specified criteria. Some other contests only require that the contestant submit the output corresponding to given input data, in which case the judge only has to analyze the submitted output data.

Online judges are online environments in which testing takes place. Online judges have ranklists showing users with the biggest number of accepted solutions and/or shortest execution time for a particular problem.

Notable competitions

There are two types of competition formats: short-term and long-term. Each round of short-term competition lasts from 1 to 5 hours. Long-term competitions can last from a few days to a few months.

- International Collegiate Programming Contest (ICPC) – one of the oldest competitions, for students of universities in groups of 3 persons each
- International Olympiad in Informatics (IOI) – one of the oldest competitions, for secondary school students

- American Computer Science League (ACSL) – computer science competition with written and programming portions, for middle/high school students
- CodeChef – competition held from 2009, there are three contests held every month and an annual competition called CodeChef SnackDown
- Codeforces Round – typically two hour contest, held every week
- Facebook Hacker Cup – competition held from 2011, provided and sponsored by Facebook
- HackerRank – multiple competitions
- Gridwars – four competitions held between 2003 and 2004.
- Google Code Jam – competition held from 2003, provided and sponsored by Google
- IEEEXtreme Programming Competition – annual competition for IEEE Student Members held since 2006 by IEEE.
- Topcoder Open (TCO) – Algorithm competition held since 2001 by Topcoder

In most of the above competitions, since the number of contestants is quite large, competitions are usually organized in several rounds. They usually require online participation in all rounds except the last, which requires onsite participation. A special exception to this is IEEEXtreme, which is a yearly 24-hour virtual programming competition. The top performers at IOI and ICPC receive gold, silver and bronze medals while in the other contests, cash prizes are awarded to the top finishers. Also hitting the top places in the score tables of such competitions may attract interest of recruiters from software and Internet companies.

The International Collegiate Programming Contest (ICPC)

The International Collegiate Programming Contest is an algorithmic programming contest for college students. Teams of three, representing their university, work to solve the most real-world problems, fostering collaboration, creativity, innovation, and the ability to perform under pressure. Through training and competition, teams challenge each other to raise the bar on the possible. Quite simply, it is the oldest, largest, and most prestigious programming contest in the world.

The ICPC traces its roots to 1970 when the first competition was hosted by pioneers of the Alpha Chapter of the UPE Computer Science Honor Society. The initiative spread quickly within the United States and Canada as an innovative program to raise increase ambition, problem-solving aptitude, and opportunities of the strongest students in the field of computing.

Over time, the contest evolved into a multi-tier competition with the first championship round conducted in 1977. Since then, the contest has expanded into a worldwide collaborative of universities hosting regional competitions that advance teams to the annual global championship round, the ICPC World Finals.

The International Collegiate Programming Contest (ICPC) is the premier global programming competition conducted by and for the world's universities. The ICPC is affiliated with the ICPC Foundation and is headquartered at Baylor University.

The contest fosters creativity, teamwork, and innovation in building new software programs, and enables students to test their ability to perform under pressure. The contest has raised aspirations and performance of generations of the world's problem solvers in the computing sciences and engineering.

The ICPC features several levels of competition:

- Local Contests
- Regional Contests
- Regional Championships
- The World Finals

Regional Rules

Registration: If a team wishes to be able to qualify for the ICPC World Finals, then before competing in ANY regional qualifying event, the coach and all team members must be fully registered in the ICPC Registration System. A qualifying contest is one that allows teams to be promoted to a higher level and which is supervised by faculty. An online contest which is unsupervised may not be classified as a qualifying contest. Registration may not be done retrospectively. Incomplete registration or circumvention that leads to incomplete or false data is grounds for immediate disqualification.

Mission: The ICPC, the "International Collegiate Programming Contest", is an extra-curricular, competitive programming sport of the universities of the world. ICPC competitions provide gifted students with opportunities to interact, demonstrate, and improve their teamwork, programming, and problem-solving prowess. The ICPC is a global platform for academia, industry, and community to shine the spotlight on and raise the aspirations of the next generation of computing professionals as they pursue excellence.

Introduction: The contest is a multi-tiered team competition for students representing institutions of higher education. The tiers range from local contests to the highest level Regional Contest from which teams advance to the ICPC World Finals. Teams have the right to advancement only for qualifying contests. The hierarchy of qualifying regional contests is shown in the ICPC Registration System. At any level, teams may be invited to advance to the next level using wildcards.

The ICPC World Finals is typically held between April and June. The date of the ICPC World Finals is published no later than August 1 before the date of the ICPC World Finals. Note: The 2022 World Finals will be held after the 2021 World Finals and announced as soon as the world situation allows.

Organization: The ICPC is organized according to the ICPC Policies and Procedures. The Executive Committee, chaired by the Executive Director, sets the policy and general rules for the conduct of the contest. The Executive Director is solely responsible for interpreting the rules and for ruling on unforeseen situations.

The ICPC is affiliated with the ICPC Foundation. The ICPC Executive Director is an officer of the ICPC Foundation which is responsible for sponsorship, fundraising, host agreements, alumni outreach, and operational matters in compliance with the ICPC Policies and Procedures within the Constitution and By-Laws of the ICPC Foundation.

The ICPC International Steering Committee is responsible for establishing regional contest rules, policies and guidelines. They oversee the conduct of regional contests, resolve regional appeals, rule on international issues, recommend ways to make the contest more accessible and attractive to international participants and review variances in regional rules. There are two standing subcommittees, the Appeals Committee, and the Eligibility Committee.

All ICPC Regional Contests are organized by the ICPC Policies and Procedures. Each Regional Contest Director (RCD) is charged with executing specific regional contests by these ICPC Regional Rules, the ICPC Policies & Procedures, and ICPC guidelines.

Each regional contest has the opportunity to vary the rules to accommodate differences in educational systems and host computing facilities. Such rules do not supersede the ICPC Regional Rules or the ICPC World Finals Rules. These contest-specific Rules and other helpful information must be posted on the regional website.

Localization: The language of the Contest is English. All written contest materials will be in English. Additional languages may be used in regional contests. Where an English term may have multiple interpretations, these must be defined to ensure all contestants have clarity of meaning.

Team Composition: A representative of the sponsoring institution of higher education, typically a faculty member, must be designated and registered as the team coach. They must certify the eligibility of contestants and be the official point-of-contact with the team before and during contest activities. A team may only have one registered coach, and that coach cannot also be a contestant.

The coach must fully register teams in the ICPC Registration System within the time set by the regional rules which, for all qualifying Regional Contests, is no later than 7 days before the contest. A team is not eligible to compete in the regional contest until the rethe Eligibility Decision Treegional contest director has accepted the team in the web registration system. Teams failing to comply with any of these requirements will be ruled ineligible to compete. Only registered reserves may be substituted for contestants. Such substitutions must be entered in the ICPC Registration System by the regional contest director before the contest begins. A reserve can only be substituted for a contestant before the first qualifying contest. Each team consists of three contestants who are

eligible to compete in the ICPC World Finals as described under Advancing to the ICPC World Finals in the Regional Rules. The team's contestants must satisfy all eligibility requirements. For more complex decisions, the Eligibility Decision Tree can be used.

Basic Eligibility Requirements: An eligible student must be willing and able to compete in the ICPC World Finals. In particular, a student who requires a visa to study at the team's institution must be able to gain a visa to travel to the ICPC World Finals plus be able to return to the institution's location after the Finals. A student must be enrolled in a degree program at the team's institution with at least a half-time load. This rule is not to be construed as disqualifying co-op students, exchange students, students serving internships, or extramural students. Note: Disruption in enrollment caused by epidemics is normally waived.

A student may compete for only one institution during a contest year.

A student who has competed in two ICPC World Finals is NOT eligible to compete. Note: ICPC World Finals Moscow participation is not counted towards the max participation in two WFs.

A student who has competed in qualifying regional contests during five different contest years is NOT eligible to compete. Note: Regional participation in 2020/21 is not counted towards the maximum of 5!

Period of Eligibility: A student who meets the Basic Requirements and FIRST began post-secondary studies in 2017 or later is eligible to compete. A student who meets the Basic Requirements and was born in 1998 or later is eligible to compete. The eligibility status is determined at the first qualifying contest. Note: If the 2021 regional year is delayed, we still determine the eligibility at the point as if the year was not delayed (normally October 1 2021).

Extending the Period of Eligibility: A coach may petition the ICPC Eligibility Committee at least three weeks before the regional contest, to extend the Period of Eligibility for a student whose full-time studies have been interrupted or extended. (This includes military or civilian service, illness, epidemics, any work, other studies, or personal reasons).

The coach must demonstrate that such an extension would not provide an unfair advantage to the team.

A petition will normally be approved if the student meets the Basic Eligibility Requirements and has not completed more than the equivalent of eight semesters of full-time STEM study as of the date of the regional contest. Note: If the 2021 regional year is delayed, we still determine the eligibility at the point as if the year was not delayed (normally October 1 2021).

The ICPC Eligibility Committee will render a decision within five days of receiving the petition.

Where to Compete: Contests are partitioned geographically into "ICPC Regions", and every institution is assigned a home ICPC Region. An institution's teams may compete in only one regional contest advancement path to the ICPC World Finals.

An institution may request to move to a different home ICPC Region for a given year for a valid reason. The team coaches from the institution must submit such a request to the Director of Regional Contests (DRC), who will approve the request only if the decision is unanimous among all affected Directors.

Guest Team Participation: An RCD director may invite guest teams who are not eligible to advance toward the World Finals, to compete in a regional contest. An eligible student may compete on a guest team in at most one additional regional contest per year. Guest teams must be registered as such before conducting the contest.

Regional Contest Attendance: All team members must attend all contest activities as specified by the regional contest director for that region. The coach is expected to attend or be available by phone or similar during contest activities. Failure to attend any of the designated contest events will result in automatic disqualification and forfeiture of any scholarships and prizes.

Regional Contest

Solutions to problems submitted for judging are called runs. Each run is judged as accepted or rejected, and the team is notified of the results.

Public notification of accepted runs may be suspended at an appropriate time (typically during the last hour of the contest) to keep the final results secret. A general announcement to that effect will be made during the contest. Notification of rejected runs will continue to be made to the team until the end of the contest.

A contestant may submit a claim of ambiguity or error in a problem statement by submitting a clarification request. If the judges agree that an ambiguity or error exists, a clarification will be issued to all contestants.

Contestants are not to converse with anyone except members of their team and personnel designated by the regional contest director. Systems support staff may advise contestants on system-related problems such as explaining system

error messages.

While the contest is scheduled for a particular time length (typically five hours), the regional contest director has the authority to alter the length of the contest in the event of unforeseen difficulties. Should the contest duration be altered, every attempt will be made to notify contestants in a timely and uniform manner.

A team may be disqualified by the regional contest director for any activity that jeopardizes the contest such as dislodging extension cords, unauthorized modification of contest materials, any activity that may knowingly or unknowingly adversely impact another team or distracting behavior.

At least six problems will be posted. So far as possible, problems will avoid dependence on detailed knowledge of a particular applications area or particular contest language.

A problem is solved when it is accepted by the judges. The judges are solely responsible for accepting or rejecting submitted runs. In consultation with the judges, the Regional Contest Director determines the winners of the regional contest. The regional contest director and judges are empowered to adjust for or adjudicate unforeseen events and conditions. Their decisions are final.

Teams are ranked according to the most problems solved. For awards, or in determining qualifier(s) for the ICPC World Finals, teams who solve the same number of problems are ranked by least total time. The total time is the sum of the time consumed for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the first accepted run plus 20 penalty minutes for every previously rejected run for that problem. There is no time consumed for a problem that is not solved.

It is the responsibility of the Regional Contest Director to specify any additional tie-breakers. Tie-breaker policies must be announced to contestants before the contest begins.

Regional Championships

The regional contests are divided into eight championship areas; the Africa & Arab Championship, the Asia East Championship, the Asia Pacific Championship, the Asia West Championship, the European Championship, the Latin America Championship, the Northern Eurasia Championship, and the North America Championship.

The highest level Regional Contest may be a Championship Contest, and thus awarding the winning team the title Champions (The Africa & Arab Champions etc). Team composition and requirements rules for a Championship contest are to be the same as for the World Finals. A Championship contest should be held no later than January 31. Note: The 2020 Regional Schedule will be able to be extended into the first quarter of 2021 - and even longer if needed. Extension beyond the first quarter of 2021 needs to be approved by HQ.

The programming languages of the regional contest will include C and C++. Additional programming languages may be used. The programming languages of the ICPC World Finals are Java, Kotlin, Python and C/C++. Before the ICPC World Finals, the judges will have solved all problems in Java and C/C++.

Each team will use a single workstation. The regional contest director is responsible for determining that teams have reasonably equivalent computing resources.

Each Regional Contest Director determines whether contestants may bring materials for use during the contest. Please see the specific regional rules at the ICPC Regional Contest Web Site. At the ICPC World Finals, no printed materials or electronic devices may be brought into the contest area. On-line reference materials will be made available as described in the Programming Environment Web Site. Each team will be permitted to provide a PDF of up to 25 pages of notes within the limits described during Team Certification. Three copies will be printed and placed at the team's workstation for use during the ICPC World Finals.

A sample task from past NERC contest

This is only a sample task. You don't need to solve it.

Daisy is a senior software engineer at RainyDay, LLC. She has just implemented three new features in their product: the first feature makes their product work, the second one makes their product fast, and the third one makes their product correct. The company encourages at least some testing of new features, so Daisy appointed her intern Demid to write some tests for the new features.

Interestingly enough, these three features pass all the tests on Demid's development server, which has index 1, but might fail the tests on some other servers.

After Demid has completed this task, Daisy appointed you to deploy these three features to all n servers of your company. For every feature f and every server s, Daisy told you whether she wants the feature f to be deployed on the server s. If she wants it to be deployed, it must be done even if the feature f fails the tests on the server s. If she does not want it to be deployed, you may not deploy it there.

Your company has two important instruments for the deployment of new features to servers: Continuous Deployment (CD) and Continuous Testing (CT). CD can be established between several pairs of servers, forming a directed graph. CT can be set up on some set of servers.

If CD is configured from the server s_1 to the server s_2 then every time s_1 receives a new feature f the system starts the following deployment process of f to s_2:

- If the feature f is already deployed on the server s_2, then nothing is done.
- Otherwise, if CT is not set up on the server s_1, then the server s_1 just deploys the feature f to the server s_2 without any testing.
- Otherwise, the server s_1 runs tests for the feature f. If the tests fail on the server s_1, nothing is done. If the tests pass, then the server s_1 deploys the feature f to the server s_2.

You are to configure the CD/CT system, and after that Demid will deploy all three features on his development server. Your CD/CT system must deploy each feature exactly to the set of servers that Daisy wants.

Your company does not have a lot of computing resources, so you can establish CD from one server to another at most 264 times.

Complaints, Appeals, and Remedies

If irregularities or misconduct are observed during the contest, team members or coaches should bring them to the attention of the contest officials so that action may be taken as soon as possible. After the conclusion of the contest and the results have been made public, coaches may file complaints or appeals as follows:

Within 1 day

The coach may file a complaint by sending an email containing a text message with no enclosures to the Contest Manager. The Contest Manager will forward the complaint to the Regional Contest Director, supervising Regional Contest Directors, and the Director of Regional Contests, copying the coach.

Within 2 more days

The RCD shall respond to the complaint.

Within 1 more day

The coach may file an appeal by sending email to the Contest Manager who will forward the appeal to the Appeals Committee copying the coach, RCD and supervising RCDs.

Within 2 more days

The Appeals Committee will investigate the circumstances of the appeal and notify the coach, RCD and the supervising RCDs.

This process is governed as follows:

The results of the regional contest are not final until the complaints and appeals process has run its course. Only coaches may file complaints and appeals. An appeal must be based on one or more of the following circumstances: violations of the Rules, misconduct by teams, or gross misconduct by contest officials with the intent to harm. The decisions of the judges are final. Specifically, a decision on a problem submission MAY NOT be appealed. The Appeals Committee overturns decisions only under extraordinary circumstances. The decision of the Appeals Committee is final. No additional finals invitations will be given to remedy a complaint. All complaints will be acknowledged.

The appeal will be automatically rejected if the above procedure is not followed.

Advancing to the ICPC World Finals

The highest level Regional Contests advance teams to the ICPC World Finals. Additional teams who competed in the highest level of regional contests may be invited to the ICPC World Finals as wild card teams.

Teams qualify to advance to the ICPC World Finals through Regional Contests and by satisfying all rules posted in The Rules of the ICPC World Finals. Specifically:

To qualify for the ICPC World Finals, the coach and all team members must be fully registered in the ICPC Registration System BEFORE competing in ANY regional qualifying event. Incomplete registration or circumvention that leads to incomplete or false data is grounds for immediate disqualification.

Only one team from a given institution may advance to the ICPC World Finals. No team member on the qualifying team may have competed as a contestant in two previous ICPC World Finals.

The coach of a qualifying team is the point-of-contact before and during ICPC World Finals activities. The coach must complete certification at the Team Certification Web Site within five (5) days of notification. Qualifying teams will be invited by email within one day of completing certification.

Qualifying teams requiring visas must initiate the process of applying for visas within ten days of being issued an invitation. Teams failing to comply with any of these requirements will be ruled ineligible to compete in the ICPC World Finals. Upon completion of these requirements, a qualifying team will be advanced to the ICPC World Finals.

A team advancing to the ICPC World Finals will be comprised of the same three members as when it qualified. If a team member is unwilling, unable or unfit to compete in the ICPC World Finals, the coach must notify the ICPC Manager promptly. A team member who is unwilling or unfit to compete in the ICPC World Finals will be disqualified from further ICPC competitions. The team member may appeal disqualification to the Appeals Committee.

At on-site registration, participants must provide a picture ID (passport, drivers license, etc). Contestants must show proof of enrollment at the university during the term of the regional contest at which they qualified. A letter on university stationery with the signature of a university official accompanied by an English translation is sufficient.

Conduct of the Finals

ICPC World Finals Championship:

Ten or more problems have been posed in recent World Finals.

Problems will be posed in English. During the contest, all communications from contest officials to contestants will be in English. Each team may identify an interpreter for translating questions posed by contestants to contest officials. Contestants may bring electronic natural language translators provided that they do not support math operations.

Solutions to problems submitted for judging are called runs. Each run is judged as accepted or rejected, and the team is notified of the results. Rejected runs will be marked as follows:

run-time error time-limit exceeded wrong answer

Notification of accepted runs may be suspended at the appropriate time to keep the final results secret. A general announcement to that effect will be made during the contest. Notification of rejected runs will continue until the end of the contest.

A contestant may submit a claim of ambiguity or error in a problem statement by submitting a clarification request. If the Judges agree that an ambiguity or error exists, a clarification will be issued to all contestants.

The competition is scheduled to last five hours. The Finals Director has the authority to lengthen the contest in the event of unforeseen circumstances. Should the Contest duration be altered, every attempt will be made to notify contestants in a timely and uniform manner.

ICPC World Finals Invitational:

The Invitational is conducted similarly to the Championship with the competition lasting five hours. The scoreboard is frozen one hour before the end of the contest. The judges will review the results, submissions, and archival materials as necessary to certify team performance. The Teams will be introduced during the World Finals Opening Ceremony. During the Championship competition, the Invitational scoreboard will be resolved to determine honors. Awards will be given at the Awards Ceremony.

Scoring of the World Finals

ICPC World Finals Championship:

The World Finals Judges are solely responsible for determining the correctness of submitted runs. In consultation with the World Finals Judges, the Director of Judging is responsible for determining the winners of the World Finals. They are empowered to adjust for or adjudicate unforeseen events and conditions. Their decisions are final.

Teams are ranked according to the most problems solved. Teams placing in the first twelve places who solve the same number of problems are ranked first by least total time and, if need be, by the earliest time of submittal of the last accepted run.

The total time is the sum of the time consumed for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the first accepted run plus 20 penalty minutes for every previously rejected run for that problem. There is no time consumed for a problem that is not solved.

ICPC World Finals Invitational:

The World Finals Judges are solely responsible for determining the correctness of submitted runs. In consultation with the World Finals Judges, the Director of Judging is responsible for determining the winners of the World Finals. They are empowered to adjust for or adjudicate unforeseen events and conditions. Their decisions are final.

For the purpose of making awards, teams are ranked according to the most problems solved. Teams placing in the top 10% who solve the same number of problems are ranked first by least total time and, if need be, by the earliest time of submittal of the last accepted run.

The total time is the sum of the time consumed for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the first accepted run plus 20 penalty minutes for every previously rejected run for that problem. There is no time consumed for a problem that is not solved.

Computing Environment

ICPC World Finals Championship:

The World Finals programming language tools include Java, C, C++, Kotlin and Python. See the Programming Environment Web Site for detailed configuration information. Before the World Finals, the judges will have solved all problems in languages from at least two of the three distinct language groups (Java/Kotlin, C/C++, and Python).

Each team member will be provided with a computer. All teams will have equivalent computing equipment. (Individual computers are provided for health safety purposes.)

Contestants may not bring any printed materials or machine-readable versions of software or data to the Contest Area. Contestants may not bring their own computers, computer terminals, calculators, or other electronic devices to the Contest Area.

Each team member may bring an unannotated natural language printed dictionary. On-line reference materials will be made available as described in the Reference Materials section of the Programming Environment Web Site. Each team will be permitted three copies of the Team Reference Document described in the On-Site Registration Instructions.

ICPC World Finals Invitational:

The World Finals programming language tools include Java, C, C++, Kotlin and Python. See the Programming Environment Web Site for detailed configuration information. Before the World Finals, the judges will have solved all problems in languages from at least two of the three distinct language groups (Java/Kotlin, C/C++, and Python).

Each team will compete at a proctored site designated by the coach and approved by an institutional representative, preferably at the institution's campus. The institution may provide each team member with a computer. All team members should have computing equipment comparable to the Championship workstations described at Programming Environment Web Site. Competition must be under the oversight of the coach and the team's institutional representative. The institutional representative must certify that the team competed within the rules, with no lapse of proctoring, in a proctored environment with video monitoring and archival materials made available to judges during the Judges' Review. Proctoring may be by recorded video, but the record must be conclusive and without interruption to qualify for ranking in the top 10%. Should a team need to compete at multiple-sites, the institutional representative must make a case for equivalence.

The sole use of Internet access is to access the on-line competition and to communicate solely with team mates.

Contestants may not bring any printed materials or machine-readable versions of software or data to their Team Area. Contestants may not bring their own computers, computer terminals, calculators, or other electronic devices to their Team Area. Each team member may bring an unannotated natural language printed dictionary. On-line reference materials will be made available as described in the Reference Materials section of the Programming Environment Web Site. Each team will be permitted three copies of the Team Reference Document described in the On-Site Registration Instructions.

The FOI Championship

The 57th **F**lower **O**lympiad in **I**nformatics Championship is based on the rules of the traditional ICPC. After the championship day, there will be a rest day followed by the Olympic competition day.

The details of the FOI Olympiad Day are based on the rules of the International Olympiad in Informatics.

What is IOI?

The International Olympiad in Informatics is one of several international science Olympiads held annually around the world. Exceptional high school students from various countries compete in the prestigious algorithmic competition to sharpen their informatics skills—such as problem analysis, design of algorithms and data structures, programming, and testing.

The contest consists of two days of computer programming/coding and problem-solving of algorithmic nature. To deal with problems involving very large amounts of data, it is necessary to have not only programmers, "but also creative coders, who can dream up what it is that the programmers need to tell the computer to do. The hard part isn't the programming, but the mathematics underneath it."Students at the IOI compete on an individual basis, with up to four students competing from each participating country (with 81 countries in 2012). Students in the national teams are selected through national computing contests, such as the British Informatics Olympiad, Indian Computing Olympiad, Romanian Olympiad in Informatics, or Bundeswettbewerb Informatik (Germany).

The International Olympiad in Informatics is one of the most prestigious computer science competitions in the world. UNESCO and IFIP are patrons.

On each of the two competition days, the students are typically given three problems which they have to solve in five hours. Each student works on his/her own, with only a computer and no other help allowed, specifically no communication with other contestants, books etc. Usually to solve a task the contestant has to write a computer program (only in C++) and submit it before the five-hour competition time ends. The program is graded by being run with secret test data. From IOI 2010, tasks are divided into subtasks with graduated difficulty, and points are awarded only when all tests for a particular subtask yield correct results, within specific time and memory limits. In some cases, the contestant's program has to interact with a secret computer library, which allows problems where the input is not fixed, but depends on the program's actions – for example in game problems. Another type of problem has known inputs which are publicly available already during the five hours of the contest. For these, the contestants have to submit an output file instead of a program, and it is up to them whether they obtain the output files by writing a program (possibly exploiting special characteristics of the input), or by hand, or by a combination of these means. Pascal will have been removed as an available programming language by 2019.

MAIN OBJECTIVES

- To discover, encourage, challenge, and recognise exceptional high school students for their talent in the field of informatics
- To foster friendly international relationships among computer scientists and informatics educators
- To bring the discipline of informatics to the attention of young people
- To promote the organisation of informatics competitions for high school students
- To encourage countries to organise future IOI competitions

Each participating country selects a team of up to four contestants to represent their nation. The team, accompanied by a team leader and deputy leader, will compete in a two-day competition. Each contestant competes individually to maximise their score by solving three algorithmic problems within five hours.

IOI 2010 for the first time had a live web scoreboard with real-time provisional results. Submissions will be scored as soon as possible during the contest, and the results posted. Contestants will be aware of their scores, but not others', and may resubmit to improve their scores. Starting from 2012, IOI has been using the Contest Management System (CMS) for developing and monitoring the contest.

No.	Year	Host	City	Contestants	Countries
33	2021	Singapore	online	351	88
32	2020	Singapore	online	343	87
31	2019	Azerbaijan	Baku	327	87
30	2018	Japan	Tsukuba	335	87
29	2017	Iran	Tehran	308*	83*
28	2016	Russia	Kazan	308	80
27	2015	Kazakhstan	Almaty	322	83
26	2014	Taiwan	Taipei	311	81
25	2013	Australia	Brisbane	299	77
24	2012	Italy	Sirmione/Montichiari	310	81
23	2011	Thailand	Pattaya	303	78
22	2010	Canada	Waterloo	297	80
21	2009	Bulgaria	Plovdiv	301	78
20	2008	Egypt	Cairo	283	78
19	2007	Croatia	Zagreb	285	77
18	2006	Mexico	Mérida	282	74
17	2005	Poland	Nowy Sacz	276	72
16	2004	Greece	Athens	291	76
15	2003	United States of America	Kenosha	265	69
14	2002	Republic of Korea	Yong-In	276	78
13	2001	Finland	Tampere	272	74
12	2000	China	Beijing	278	72
11	1999	Turkey	Antalya-Belek	253	65
10	1998	Portugal	Setúbal	241	68
9	1997	South Africa	Cape Town	221	63
8	1996	Hungary	Veszprém	220	57
7	1995	Netherlands	Eindhoven	210	51
6	1994	Sweden	Haninge	189	49
5	1993	Argentina	Mendoza	155	43
4	1992	Germany	Bonn	171	51
3	1991	Greece	Athens	68	23
2	1990	Soviet Union	Minsk	100	25
1	1989	Bulgaria	Pravetz	46	13

A sample task from past IOI

This is only a sample task. You don't need to solve it.

According to ancient Persian legends in Shahnameh, Zal, the legendary Persian hero, is madly in love with Rudaba, the princess of Kabul. When Zal asked for Rudaba's hand in marriage, her father gave him a challenge.

In Persia there are n cities, labeled from 0 to $n-1$, and m two-way roads, labeled from 0 to $m-1$. Each road connects a pair of distinct cities. Each pair of cities is connected by at most one road. Some of the roads are *royal roads* used for travels by royals. Zal's task is to determine which of the roads are the royal roads.

Zal has a map with all the cities and the roads in Persia. He does not know which of the roads are royal, but he can get help from Simurgh, the benevolent mythical bird who is Zal's protector. However, Simurgh does not want to reveal the set of royal roads directly. Instead, she tells Zal that the set of all royal roads is a *golden set*. A set of roads is a golden set if and only if:

- it has **exactly** $n-1$ roads, and

- for every pair of cities, it is possible to reach one from the other by traveling only along the roads of this set.

Furthermore, Zal can ask Simurgh some questions. For each question:

1. Zal chooses a **golden** set of roads, and then
2. Simurgh tells Zal how many of the roads in the chosen golden set are royal roads.

Your program should help Zal find the set of royal roads by asking Simurgh at most q questions. The grader will play the role of Simurgh.

The DOMjudge System

The **F**lower **O**lympiad in **I**nformatics Championship will use the DOMjudge system. DOMjudge is an automated judge system to run programming contests. It has a mechanism to submit problem solutions, have them judged fully automatically and provides (web)interfaces for teams, the jury and the general public.

DOMjudge is primarily focused to be used in programming contests like the ICPC programming contests, where teams are on-site and there is a fixed problem set and time frame. It can however also be adapted to other contexts.

The system scales well: distributed judging is easy and it has a modular system for plugging in languages or compilers. The team interface is kept simple and efficient, while the administrator's view has many features, including rejudging, clarifications, detailed submission/judging information. There's a REST API to extend DOMjudge in any direction you need.

The DOMjudge system has been used in many live contests, a selection:

- ICPC Sub-regionals: BAPC (since 2004), GCPC (since 2010), UKIEPC (since 2015)
- ICPC Regionals: NWERC (since 2007), SWERC (since 2008), SER USA (since 2010), SOCAL (since 2013), RMRC (since 2014), South Pacific Regional
- ICPC World Finals (since 2012, in 2018 and 2019 as main system)

DOMjudge has also been adapted for use in numerous online contests, as "courseware"in teaching environments, etc. Only a few links (some might be dead): online judge at FAU, courseware at UU, Programa-Me (a program for high school students), IIT Kanpur "Chaos"contest. If you are using (an adapted version of) DOMjudge, we would be glad to hear about it, and very grateful for your feedback.

DOMjudge requires the following to be available to run. Please refer to the DOMserver and Judgehost chapters for detailed software requirements.

- At least one machine to act as the DOMjudge server (or domserver for brevity). The machine needs to be running Linux (or possibly a Unix variant) and a webserver with PHP 7.2.5 or newer. A MySQL or MariaDB database is also needed.
- A number of machines to act as judgehosts (at least one). They need to run Linux with (sudo) root access. Required software is the PHP commandline client and compilers for the languages you want to support.
- Team workstations, one for each team. They require only a modern web browser to interface with DOMjudge, but of course need a local development environment for teams to develop and test solutions. Optionally these have the DOMjudge submit client installed.
- Jury / admin workstations. The jury members (persons) that want to configure and monitor the contest need just any workstation with a web browser to access the web interface. No DOMjudge software runs on these machines.

One (virtual) machine is required to run the DOMserver. The minimum amount of judgehosts is also one, but preferably more: depending on configured timelimits, and the amount of testcases per problem, judging one solution can tie up a judgehost for several minutes, and if there's a problem with one judgehost it can be resolved while judging continues on the others.

As a rule of thumb, we recommend one judgehost per 20 teams.

However, overprovisioning does not hurt: DOMjudge scales easily in the number of judgehosts, so if hardware is available, by all means use it. But running a contest with fewer machines will equally work well, only the waiting time for teams to receive an answer may increase.

Each judgehost should be a dedicated (virtual) machine that performs no other tasks. For example, although running a judgehost on the same machine as the domserver is possible, it's not recommended except for testing purposes. Judgehosts should also not double as local workstations for jury members. Having all judgehosts be of uniform hardware configuration helps in creating a fair, reproducible setup; in the ideal case they are run on the same type of machines that the teams use.

DOMjudge supports running multiple judgedaemons in parallel on a single judgehost machine. This might be useful on multi-core machines.

Configuring the DOMjudge system

Configuration of the judge system is done by logging in as administrator to the web interface. The default username is `admin` with initial password stored in `etc/initial_admin_password.secret`.

The general system settings can be accessed under Configuration settings. Changes take effect immediately.

Setting up users and teams: Under Users from the homepage you can add user accounts for the people accessing your system. There are several roles possible:

- Administrative user: can configure and change everything in DOMjudge.
- Jury user: can view submissions and judgings. Can view clarification requests and send clarifications. Can rejudge non-correct judgings (submissions judged correct can only be rejudged by an administrator).
- Balloon runner: can only view the balloon queue and mark balloons as delivered.
- Team member: can view its own team interface and submit solutions (see below).
- Several system roles: they are for API access. The most important one is judgehost which means the account credentials can be used by a judgedaemon.

Administrative user: can configure and change everything in DOMjudge.

Jury user: can view submissions and judgings. Can view clarification requests and send clarifications. Can rejudge non-correct judgings (submissions judged correct can only be rejudged by an administrator).

Balloon runner: can only view the balloon queue and mark balloons as delivered.

Team member: can view its own team interface and submit solutions (see below).

Several system roles: they are for API access. The most important one is judgehost which means the account credentials can be used by a judgedaemon.

To set up teams, you can start in the Teams page and add teams there. You then have the option to automatically create a corresponding user account that is associated with the team.

It is also possible to use the *Import / Export* page to import ICPC-compatible teams.tsv files with teams.

A jury or administrative user can also be associated with a team. This will enable that user to submit solutions to the system, or resubmit edited team solutions.

Resetting the password for a user

If you do not have access anymore to any admin user, you can use the following command to reset the password of a user to a random value:

```
webapp/bin/console domjudge:reset-user-password admin
```

Replace admin with the username of the user you want to reset the password for. The password will be displayed.

Adding a contest

You configure a new contest by adding it under the Contests link from the main page.

Besides the name the most important configuration about a contest are the various time milestones.

A contest can be selected for viewing after its activation time, but the scoreboard will only become visible to public and teams once the contest starts. Thus no data such as problems and teams is revealed before then.

When the contest ends, the scores will remain displayed until the deactivation time passes.

DOMjudge has the option to 'freeze' the public and team scoreboards at some point during the contest. This means that scores are no longer updated and remain to be displayed as they were at the time of the freeze. This is often done to keep the last hour interesting for all. The scoreboard freeze time can be set with the freezetime milestone.

The scoreboard freezing works by looking at the time a submission is made. Therefore it's possible that submissions from (just) before the freezetime but judged after it can still cause updates to the public scoreboard. A rejudging during the freeze may also cause such updates. The jury interface will however always show the actual scoreboard.

Once the contest is over, the scores are not directly 'unfrozen'. You can release the final scores to team and public interfaces when the time is right. You can do this either by setting a predefined unfreezetime in the contest table, or you push the 'unfreeze now' button in the jury web interface, under contests.

All events happen at the first moment of the defined time. That is: for a contest with starttime "12:00:00" and endtime "17:00:00", the first submission will be accepted at 12:00:00 and the last one at 16:59:59.

Setting up problems

When this is done, you can upload the intended problems that teams need to solve under Problems. DOMjudge supports uploading them as *a zip file* or configuring each problem manually via the interface. You can add a problem to a contest while uploading, or associate it by editing the contest from the Contests page later.

It is possible to change whether teams can submit solutions for that problem (using the toggle switch 'allow submit'). If disallowed, submissions for that problem will be rejected, but more importantly, teams will not see that problem on the scoreboard. Disallow judge will make DOMjudge accept submissions, but leave them queued; this is useful in case an unexpected problem shows up with one of the problems. Timelimit is the maximum number of seconds a submission for this problem is allowed to run before a 'TIMELIMIT' response is given (to be multiplied possibly by a language factor). A 'timelimit overshoot' can be configured to let submissions run a bit longer. Although DOMjudge will use the actual limit to determine the verdict, this allows judges to see if a submission is close to the timelimit.

Problems can have special compare and run scripts associated to them, to deal with problem statements that require non-standard evaluation.

From the front page the Config checker is available. This tool will do a basic check of your DOMjudge setup and gives helpful hints to improve it. Be sure to run it when you've set up your contest.

Testing jury solutions

Before a contest, you will want to have tested your reference solutions on the system to see whether those are judged as expected and maybe use their runtimes to set timelimits for the problems.

The simplest way to do this is to include the jury solutions in a problem zip file and upload this. You can also upload a zip file containing just solutions to an existing problem. The zip archive has to adhere to the problem format. For this to work, the jury/admin user who uploads the problem has to have an associated team to which the solutions will be assigned. The solutions will automatically be judged if the contest is active (but it need not have started yet). You can verify whether the submissions gave the expected answer in the Judging Verifier, available from the jury index page.

Running the contest

Team Status: Under the Teams menu option, you can get a general impression of the status of each team: a traffic light will show either of the following:

- gray: the team has not (yet) connected to the web interface at all;
- red: he team has connected but not submitted anything yet;
- yellow: one or more submissions have been made, but none correct;
- green: the team has made at least one submission that has been judged as correct.

This is especially useful during the practice session, where it is expected that every team can make at least one correct submission. A team with any other colour than green near the end of the session might be having difficulties.

Clarification Requests: Communication between teams and jury happens through Clarification Requests. Everything related to that is handled under the Clarifications menu item.

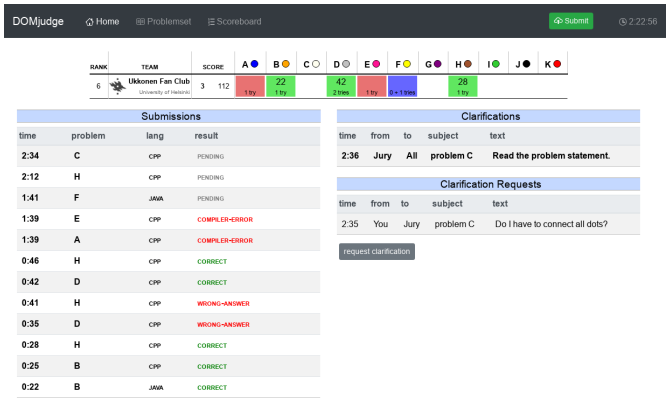
Teams can use their web interface to send a clarification request to the jury. The jury can send a response to that team specifically, or send it to all teams. The latter is done to ensure that all teams have the same information about the problem set. The jury can also send a clarification that does not correspond to a specific request. These will be termed general clarification.

Handling clarification requests: Under Clarifications, three lists are shown: new clarifications, answered clarifications and general clarifications. Click the excerpt for details about that clarification request.

Every incoming clarification request will initially be marked as unanswered. The menu bar shows the number of unanswered requests. A request will be marked as answered when a response has been sent. Additionally it's possible to mark a clarification request as answered with the button that can be found when viewing the request. The latter can be used when the request has been dealt with in some other way, for example by sending a general message to all teams.

An answer to a clarification request is made by putting the text in the input box under the request text. The original text is quoted. You can choose to either send it to the team that requested the clarification, or to all teams. In the latter case, make sure you phrase it in such a way that the message is self-contained (e.g. by keeping the quoted text), since the other teams cannot view the original request.

In the DOMjudge configuration under `clar_answers` you can set predefined clarification responses that can be selected when processing incoming clarifications.



Clarification categories and queues: When sending a clarification request, the team needs to select an appropriate category (or subject). DOMjudge will generate a category for every problem in every active contest. You can define additional categories in the DOMjudge configuration under `clar_categories`.

Categories are hence visible to the teams and they need to select one. In addition to this there's the concept of queues. Queues are purely internal to the jury, not visible to the outside world and can be used for internal workload assignment. In the DOMjudge configuration you can define in `clar_queues` the available queues and a `clar_default_problem_queue` where newly created requests will end up in. On the clarification overview page, you can quickly assign incoming clarifications to a queue by pressing the queue's button in the table row.

Balloon handling: According to ICPC tradition, every solved problem earns the team a balloon in colour specific to that problem. This can be facilitated with the balloons interface reachable from the main menu.

The interface can be accessed by administrators and any user with the Balloon runner role. It's possible to assign a user only this role; they will be able to access the jury interface but only see and handle balloons. You need to enable balloons processing for a contest in the configuration under the Contests menu option.

For each first correct submission of a problem by a team, an entry is created in the table on the balloons interface. A runner can then be dispatched to hand out the inflatable award. The entry can be marked as 'done' by clicking the running person at the end of the row.

Each row will also list the total amount of balloons that team has earned so the runner can compare the resulting situation at the team's workplace with the expected one. Where applicable there's also an indication of whether this balloon is the first in the entire contest, or the first for that problem to be handed out, should you wish to do something special with these cases.

Normally balloon distribution stops when the scoreboard is frozen. This will be indicated in the interface and no new entries will show for submissions after the freeze. It is possible that new entries appear for some times after the freeze, if the result of a submission before the freeze is only known after (this can also happen in case of a *Rejudging*). The global configuration option **show_balloons_postfreeze** will ignore a contest freeze for purposes of balloons and new correct submissions will trigger a balloon entry in the table.

Flow of a submission: The flow of an incoming submission is as follows.

1. Team submits solution. It will either be rejected after basic checks, or accepted and stored as a submission.
2. The first available judgehost compiles, runs and checks the submission. The outcome and outputs are stored as a judging of this submission. Note that judgehosts may be restricted to certain contests, languages and

problems, so that it can be the case that a judgehost is available, but not judging an available submission.

3. If verification is not required, the result is automatically recorded and the team can view the result and the scoreboard is updated (unless after the scoreboard freeze). A judge can optionally inspect the submission and judging and mark it verified.

4. If verification is required, a judge inspects the judging. Only after it has been approved (marked as verified) will the result be visible outside the jury interface. This option can be enabled by setting `verification_required` on the configuration settings admin page.

Rejudging: In some situations it is necessary to rejudge one or more submissions. This means that the submission will re-enter the flow as if it had not been judged before. The submittime will be the original time, but the program will be compiled, run and tested again.

This can be useful when there was some kind of problem: a compiler that was broken and later fixed, or testdata that was incorrect and later changed. When a submission is rejudged, the old judging data is kept but marked as invalid.

You can rejudge a single submission by pressing the 'Rejudge' button when viewing the submission details. It is also possible to rejudge all submissions for a given language, problem, team or judgehost; to do so, go to the page of the respective language, problem, team or judgehost, press the 'Rejudge all' button and confirm.

There are two different ways to run a rejudging, depending on whether the create rejudging button is enabled:

- Without this button toggled, an instant rejudging is performed where the results are directly made effective.
- When toggled, a "rejudging" set is created, and all affected submissions are rejudged, but the new judgings are not made effective yet. Instead, the jury can inspect the results of the rejudging (under the rejudging tab). Based on that the whole rejudging, as a set, can be applied or cancelled, keeping the old judgings as is.

Submissions that have been marked as 'CORRECT' will not be rejudged. Only DOMjudge admins can override this restriction using a tickbox.

Teams will not get explicit notifications of rejudgings, other than a potentially changed outcome of their submissions. It might be desirable to combine rejudging with a clarification to the team or all teams explaining what has been rejudged and why.

Ignoring a submission: There is the option to ignore specific submissions using the button on the submission page. When a submission is being ignored, it is as if was never submitted: it will show strike-through in the jury's and affected team's submission list, and it is not visible on the scoreboard. This can be used to effectively delete a submission for some reason, e.g. when a team erroneously sent it for the wrong problem. The submission can also be unignored again.

Enforcement of time limits: Time limits within DOMjudge are enforced primarily in CPU time, and secondly a more lax wall clock time limit is used to make sure that submissions cannot idle and hog judgedaemons. The way that time limits are calculated and passed through the system involves a number of steps.

Time limits are set per problem in seconds. Each language in turn may define a time factor (defaulting to 1) that multiplies it to get a specific time limit for that problem/language combination. This is the 'soft timelimit'. The configuration setting timelimit overshoot is then used to calculate a 'hard timelimit'. This overshoot can be specified in terms of an absolute and relative margin.

The soft:hard timelimit pair is passed to runguard, the wrapper program that applies restrictions to submissions when they are being run, as both wall clock and CPU limit. This is used by runguard when reporting whether the soft, actual timelimit has been surpassed. The submitted program gets killed when either the hard wall clock or CPU time has passed.

How are submissions being judged?

The DOMjudge contest control system is fully automated. Judging is done in the following way:

Submitting solutions: With the submit program or the web interface (see the section on submitting) you can submit a solution to a problem to the judges. Note that you have to submit the source code of your program (and not a compiled program or the output of your program).

On the contest control system your program enters a queue, awaiting compilation, execution and testing on one of the autojudges.

Compilation: Your program will be compiled on an autojudge machine running Linux. All submitted source files will be passed to the compiler which generates a single program to run. For Java and Kotlin the given main class will be checked; for Python we do a syntax check using the `py_compile` module.

Testing: After your program has compiled successfully it will be executed and its output compared to the output of the judges. Before comparing the output, the exit status of your program is checked: if your program exits with a non-zero exit code, the result will be a run-error even if the output of the program is correct! There are some restrictions during execution. If your program violates these it will also be aborted with a run-error, see the section on restrictions.

When comparing program output, it has to exactly match to output of the judges, except that some extra whitespace may be ignored (this depends on the system configuration of the problems). So take care that you follow the output specifications. In case of problem statements which do not have unique output (e.g. with floating point answers), the system may use a modified comparison function. This will be documented in the problem description.

Restrictions: Submissions are run in a sandbox to prevent abuse, keep the jury system stable and give everyone clear and equal environments. There are some restrictions to which all submissions are subjected:

- compile time
  - Compilation of your program may take no longer than 30 seconds. After that, compilation will be aborted and the result will be a compile error. In practice this should never give rise to problems. Should this happen to a normal program, please inform the judges right away.
- source size
  - The total amount of source code in a single submission may not exceed 256 kilobytes, otherwise your submission will be rejected.
- memory
  - The judges will specify how much memory you have available during execution of your program. This may vary per problem. It is the total amount of memory (including program code, statically and dynamically defined variables, stack, Java/Python VM, ...)! If your program tries to use more memory, it will most likely abort, resulting in a run error.
- creating new files
  - Do not create new files. The sandbox will not allow this and the file open function will return a failure. Using the file without handling this error can result in a runtime error depending on the submission language.
- number of processes
  - You are not supposed to explicitly create multiple processes (threads). This is to no avail anyway, because your program has exactly 1 processor core fully at its disposal. DOMjudge executes submissions in a sandbox where a maximum of 64 processes can be run simultaneously (including processes that started your program).
  - People who have never programmed with multiple processes (or have never heard of "threads") do not have to worry: a normal program runs in one process.

Quite Easily Done!

That's all the technical details we need to know. Now it's time to prepare a contest!

After reading these technical details about the game, Little Q felt so confused and dizzy. She even forgot what she was doing! Please help her to recall the name of the contest!

Input

There's no input for this problem.

Output

Output a single line, contains the name of the contest that Little Q needs to prepare.

Examples

standard input	*standard output*
	National Olympiad in Sleeping
	I Can't Program Carefully
	If you are a man, you have to solve eight problems

Notes

Note the sample output is wrong.