acm icpc International Collegiate Programming Contest

icpc global programming tools sponsor

South Pacific Region

ACM SOUTH PACIFIC DIVISIONAL PROGRAMMING CONTEST

Welcome to the 2018 South Pacific ACM Divisional Programming Contest, sponsored this year by **Jet Brains**. Every year this competition gathers university teams from the South Pacific Region to compete in solving problems posed by a team of judges. The 2018 ACM South Pacific Regional Finals will be held in Sydney on 23rd to 25th November this year. The Regional Finals will be used to find our 2 top teams who will be invited to participate in the ACM ICPC World Finals in 2018.

Qualification for the Regional Finals is as follows:

- **Selection Step I**: The top 2 universities from each Division will qualify for the Regional Finals. (6 teams total will advance this way)
- **Selection Step II**: After removing the teams that qualified in Selection Step I from the scoreboard, a further 6 teams are selected from the scoreboard in rank order, excluding any teams that would breach the constraint that at most 2 teams from any one University can compete in the Regional Finals.
- **Selection Step III**: Further spot(s) may be awarded via a wildcard position. This will depend on participation in the ANZAC league.

In order to make the day as enjoyable as possible, please keep to the rules and follow the guidelines carefully.

GENERAL DETAILS

The competition will run for 5 hours. There are a number of problems to solve, one computer for each team of three, and shared printing facilities.

1 General instructions

Please note these important issues relating to the actual running of the competition.

⇒ Your contest pack should identify your team by number and name.

⇒ We have 8 or more problems for the 2018 Contest. The difficulty of the problems cannot be inferred from the order of the problem set.

⇒ Your solutions are judged by comparing your program's output to our correct output. Therefore, you *must stick exactly to the output format* described in the problem statement, including all punctuation and letter cases.

⇒ From time to time teams will find a problem statement difficult to understand or confusing, and will request a clarification. If the issue is deemed by the judges to be clearly specified in the problem statement, a feedback message telling the team to read the statement and/or the sample data more carefully will be

returned only to the requesting team. If the clarification is deemed to be due to an ambiguity or an imprecision in the problem statement, a clarification message will be sent to all the competing teams in all sites

2 Conduct during the competition: dos and don'ts

⇒ After the practice session, teams may not touch their allocated computer, nor the problem set, until the start of the contest. This includes opening text editors/IDEs, setting up directories, logging into DOMjudge, etc.

Note that if there is a practice contest beforehand, all your files will be deleted before the main contest.

⇒ All accesses from a team's workstation are monitored; any attempt at unauthorised access such as email will result in instant disqualification. Using the internet in any way is not permitted, except as required for submitting solutions to be judged.

⇒ Teams are expected to work quietly and avoid being disruptive. The judges reserve the right to disqualify a team in any way disrupting the running of the competition.

⇒ Calculators, mobile phones and other electronic devices are not allowed. These should be left with your coach or a contest official.

⇒ Contestants may use any printed material, including books, but must not bring any machine-readable information into the lab. Please leave all memory devices with your coach or a contest official.

3 Submissions and feedback

⇒ Your solutions may be written in one of the available languages.

⇒ For the main contest, the languages are Java, C/C++ and Python 3. If you submit a solution in any other language (C#, for example) although it will be judged, and you will appear on the scoreboard and may win local prizes, you will NOT be able to qualify for the Regional Finals. This is true even if your submission is judged incorrect and you gain no points for it.

⇒ The judges will have implemented a solution for each problem in C/C++ and Java, but it is not guaranteed that a solution exists in Python for any of the problems since Python may be too slow. The time limit for a problem will be the same no matter which language you use.

⇒ School teams and other unofficial teams may use any available language.

⇒ Submissions are sent to the judges through DOMjudge. A separate set of instructions for using this competition software is provided.

⇒ For all languages, all code to solve a problem must be included in a single source file. If you use any file extensions apart from the standard ones for the language you are using (eg .java or .cpp), make sure you select the correct language from the Language drop down list on the DOMjudge submission page.

⇒ Make sure you select the correct problem from the Problem drop down list on the DOMjudge submission page.

⇒ You will be informed by the judging system of the result of your submission:

- A successful submission will return a `'CORRECT'` message, and your team will have solved one more problem. The time penalty will be the number of minutes into the contest at which the submission was made, plus 20 minutes for any rejected submission for this problem.

- A failed submission will be returned with an error message (see separate DOMjudge Team Manual section 2.1). You may edit your code and resubmit as many times as you like, but each rejected solution will cost you a 20 minute time penalty if you eventually solve the problem.

⇒ It is possible that a conceptually correct solution will exceed the limits of the machine used for judging, for example will run out of the default stack space, and generate a run-time exception. Please note that all problems have solutions that work with the default configurations. You may wish to take this into account if you keep getting inexplicable run-time errors.

⇒ Please remember that the judges are testing your solution with more comprehensive data than the sample data provided in the problem set. A solution may pass with the sample data, but fail with the judges' data because of test cases you have not considered. It is up to you to produce your own test data to help locate errors that the judges found.

TECHNICAL DETAILS

4 Input/Output Format

⇒ All input and output files use the ASCII encoding, 1 character per byte. They use only 7–bit ASCII characters in the printable range space (‘ ’) to tilde (‘~’), plus end–of–lines terminators (more details in the END–OF–LINES section) – tabs (‘`\t`’) should not be used.

⇒ Ignore any trailing spaces at the end of lines and trailing blank lines at the end of files (if any). Correspondingly, the judging system will ignore any such trailing spaces or blank lines.

5 End–Of–Lines

⇒ Judging will be done on a Unix system, so only Unix LF ('\n') will be the end-of-line terminators on the data files. Teams on other platforms should write code that doesn't process input at the byte level (instead use higher-level input methods such as cin, readline or Scanner).

6 Reading from Input and Writing to Output

⇒ You will be required to read from a redirected ('<') standard input stream (i.e., use **System.in**, **stdin**, **cin**, **Console.in**, or equivalent) and write to a redirected ('>') standard output stream (i.e., use **System.out**, **stdout**, **cout**, **Console.out**, or equivalent – never write your output to the standard error stream). Hypothetical example:

```
P1 <  P1_in_1.txt  >  MyOutput.txt
```

7 Judging Environment

⇒ All submissions will be judged on the DOMjudge server at University of Canterbury. When writing your solutions bear in mind that they will be judged using the following:

- Java 1.8.0
- gcc 6.3.0
- Python 3.5.3
- Mono gmcs C# 4.0 (NOT an official contest language).

⇒ Make sure you do not include IDE specific libraries which will not compile on the judging machine.

8 Site Specific Information

Each site will provide contestants with information required to login and use the local system.