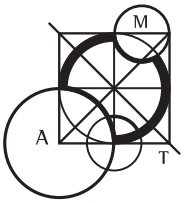
Australian Mathematics Trust



Australian Informatics Olympiad
Thursday 31 August, 2017

Information Booklet

Information for Teachers and Students

Contest Rules

Why Did I Score Zero?

Please read this booklet before the day of the contest

Information for Teachers and Students

Please read through this information carefully. If you have any questions, you are encouraged to email the judges at *aioquery@amt.edu.au*. We will endeavour to answer any queries as soon as possible.

Contest System

The contest will be run through the contest system: `http://aio.edu.au/contest`

Before the Contest

- Please ensure that students and teachers are familiar with the contest rules. Note that students may not see the actual contest questions until the beginning of the three-hour contest period.
- **Teachers will need to register their schools and students beforehand**. As there is a practice contest available, we recommend you register as soon as possible. You can register your school through the AIO registration site: `http://aio.edu.au/register`
  - During registration, we will ask for contact details for the supervising teacher. This will allow us to contact you if anything goes wrong.
  - We will also ask for an email address for each student. This will allow us to contact the student in the event that they perform very well and are invited to the AIOC School of Excellence in December.
  - Upon registering your school, you will receive a username and password for each student. Give these to each student so they can practise. This will allow students to submit their own solutions during the contest.
- Students will have the chance to attempt a practice contest run on the same system as the AIO.
  - The practice contest will be made available approximately a week before the AIO at the AIO practice site: `http://aio.edu.au/practice`, and will be available until the end of the contest.
  - Once they are registered through the AIO registration site, students will have the chance to practise using the contest system.
  - Sample problems and solution templates will be available for download through the system and solutions to these problems can be submitted for automatic marking and feedback. The practice contest problems will be different from the actual AIO problems.
  - Unlike the AIO, students will not be limited to a three-hour block to complete the practice contest. They are free to practise using the system up until the start of the AIO.
  - If students have any questions or queries regarding the practice contest or the contest system, they should send an email to *aioquery@amt.edu.au*.
  - Students will not receive any credit for participating and/or solving problems in the practice contest.

Starting the Contest

- Teachers should give each student their question booklet, along with the student's username and password from registration.
- To start the contest, teachers should instruct students to log in to the contest system with their username and password and click the red Start button to start their contest timer. At this point they may advise students to open their question booklets and begin working.

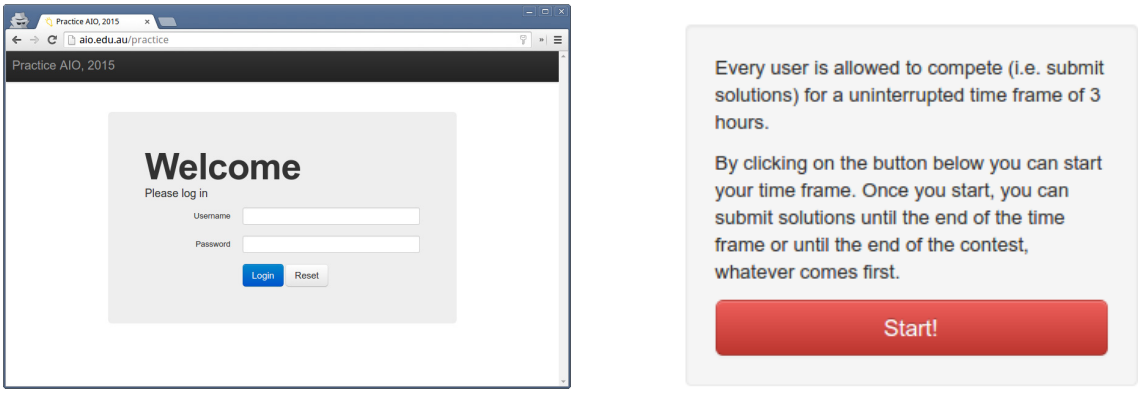


Figure 1: Login screen and start button.

During the Contest

- Students will be able to download an electronic copy of each question and its solution templates from the respective *Statement* tabs.

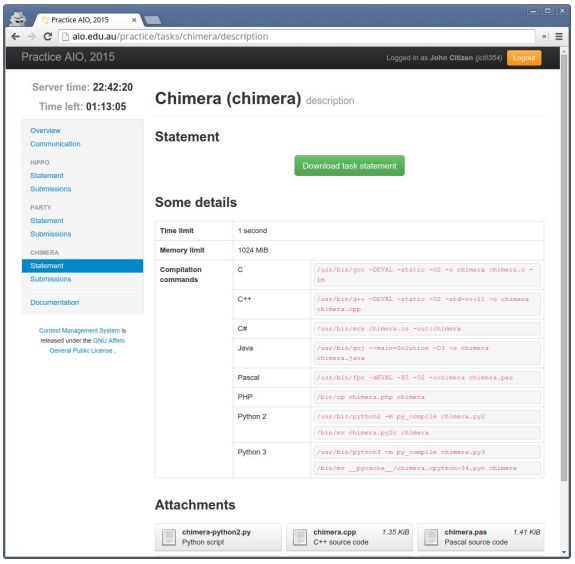


Figure 2: *Statement* tab with question statement (top) and solution templates (bottom) available for download.

- Students should submit their solutions through this system **during the three-hour contest period**.

- Once the contest system receives a submission, it will automatically judge it against the judges' input data (a set of **secret** test cases) and will award a score. Since this process may take some time, students are advised to **be patient and continue working**.

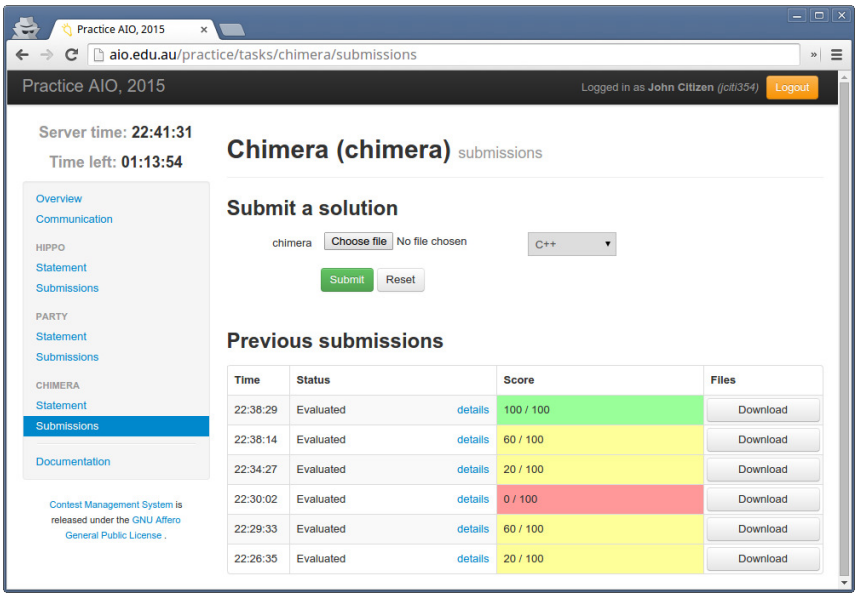


Figure 3: The *Submissions* tab allows students to submit problems by selecting a file and their language. It also displays a score summary of a student's submissions to a problem; only the **highest** scoring submission counts. Students are also able to download their code from previous submissions here.

- Students will be able to view the score of each of their submissions through the contest system. They can click the *details* button for a breakdown of the score for a submission.
- The *details* tab also displays any output from the compiler (where applicable), **including warnings and/or compilation errors**.

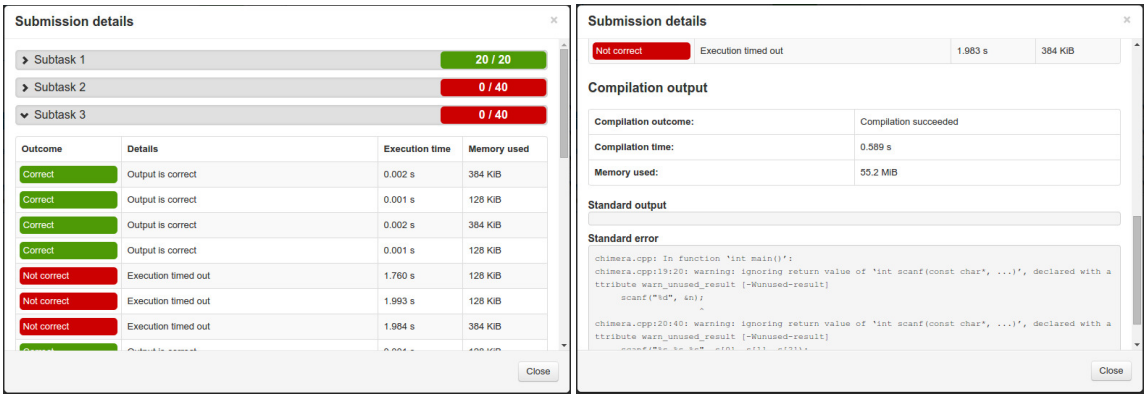


Figure 4: *details* tab, showing the score breakdown and compiler output of a student's submission.

- For further details on how scores are calculated, see the *Judging* section on page 9.
- Students may submit at most once per minute to each question. Their score for each question will be the **maximum** among all of their submissions to that question.

- **Students will be unable to make any submissions once the three hours are over!** Please encourage your students not to leave all their submissions until the last few minutes—otherwise they risk running out of time. Students are advised to submit each solution once it is written. (Remember, they may always resubmit a better solution later.)
- The solution for each problem should be the source code for a single computer program.
- When submitting solutions, students should submit the actual source code (such as *file.c*), not the compiled executable (such as *file.exe*). **Compiled executables will result in a 'Compilation failed' verdict, and will receive a score of zero.**
- If students have any queries regarding the contest questions, they should send a message to the judges using the *Communication* tab in the contest system. All announcements and clarifications regarding the contest will be posted there.
  - The judges will answer questions about the contest system, e.g. 'The system will not accept any more of my submissions, what's wrong?'
  - The judges will also answer **Yes/No questions** about the tasks where they deem a question's statement to be ambiguous. In this case, they will respond with either 'Yes', 'No', 'Answered in Task Description', 'Invalid Question' or 'No comment'.
  - The judges may, at their discretion, ignore questions that are considered unrelated or solely argumentative.

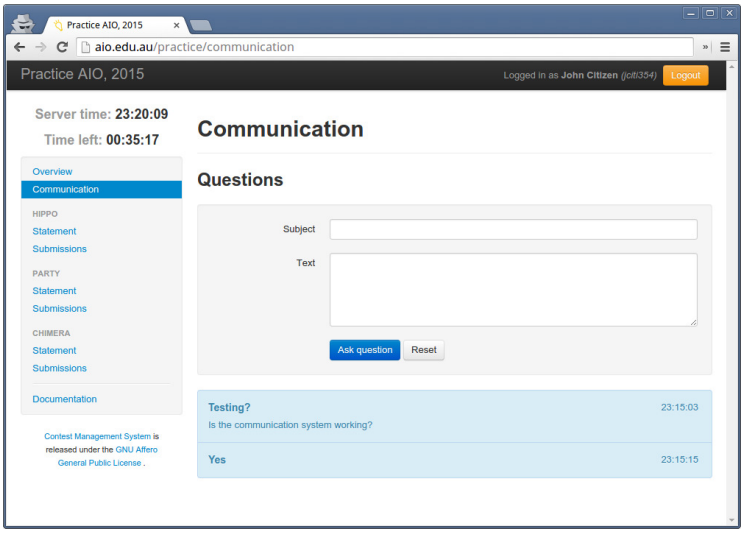


Figure 5: The *Communication* tab allows students to ask the judges questions.

After the Contest

- If students were unable to submit solutions for any reason, teachers should send one email per student to *aioquery@amt.edu.au* at the end of the contest. Each email should state the reason why the student was unable to submit during the contest (e.g. network problems), the student's name and username and have the student's code for each problem attached. Any additional points gained from these submissions may be added to the student's score, at the judges' discretion. Please note that the results of these submissions will not be revealed until results are finalised. **Absolutely no submissions will be accepted after 11:59 pm on Friday 1 September 2017.**

- Once results are finalised by the judges, students will be able to log back into the contest system to see their scores, as well as written feedback from the judges on their solution and how it might be improved, where appropriate.

Queries and Difficulties

- If you (the teacher) have any questions regarding the contest, please contact the judges by email at *aioquery@amt.edu.au*. Your query will be answered as soon as possible.
- For urgent problems on the day of the contest (such as not being able to log in or errors in the contest system), please contact either Mr Robert Newey on 0432 748 904 or Mr Joshua Lau on 0466 964 264.

Contest Rules

The contest rules are set by the AIOC Problems Committee. This committee alone is responsible for the interpretation of the rules and of the contest questions, and is fully responsible for clarifying or altering the rules or contest questions in unforeseen circumstances.

Please read these rules well before the contest. If you have any queries regarding the conduct of the contest, please email your query before the contest to *aioquery@amt.edu.au*. Your query will be answered as soon as possible.

Eligibility

- All students currently enrolled in an Australian secondary school (or equivalent overseas institution) are eligible to enter the 2017 Australian Informatics Olympiad.

Contest Duration and Start Time

- The contest will last for three hours, to be held in a single block. All students at the school must sit the contest at the same time.
- The contest must begin between 9 am and 2 pm inclusive, Australian Eastern Standard Time, on Thursday 31 August 2017. For instance, the contest may be held from 1 pm till 4 pm, but not from 3 pm till 6 pm.

Contest Environment

- Each student should have access to one and only one computer.
- Students may, if desired, use the following resources:
  - C, C++, C#, Pascal or Java compilers, PHP or Python interpreters, debuggers and associated IDEs
  - the solution templates provided from the submission website
  - calculators and printers
  - any books or other written material, including printed source code
  - any documentation in electronic form, such as help files—this does *not* include electronic source code
  - the contest system `http://aio.edu.au/contest`, for submitting solutions, communicating with the judges or downloading electronic versions of contest material
  - other people for administrative matters (e.g. *How much time do I have left? Are we allowed to use calculators? Why is there smoke coming out of my monitor?*)

- Students may *not* use the following resources during the contest:
  - compilers or interpreters for any computer language other than those listed above
  - the internet, for any purpose other than accessing the contest system as described above
  - any source code in electronic form (either your own code, sample code that comes with your compiler or non-standard libraries that come with your compiler), with the exception of the code templates provided
  - other people for technical matters (e.g. *How do you program a* for *loop in Pascal? What does this paragraph in the question mean? How many bytes does an integer take up?*) Students can send a message using the *Communication* tab in the contest system, or teachers can email the question to *aioquery@amt.edu.au*.
- Students may not communicate with other contestants.

Program Restrictions

- Students should write a computer program to solve each problem.
- Programs should read input only from the input file(s) specified in the question statements, and should send output only to the output file(s) specified in the question statements. The input and output files should be assumed to be in the current directory. **Any output to the screen will be ignored, and no input from the keyboard will be supplied.**
- The format of the input file will be specified in each problem statement.
- The desired format of the output file will also be specified in each problem statement. If you do not adhere to this output format, you may lose marks for your solution. The only exception to this will be that judges will ignore any spaces at the beginning and end of each output line, unless otherwise specified in the problem statement.
- Each solution should be a *single* source file, written in one of the following languages:
  - C
  - C++
  - C#
  - Pascal
  - Java
  - PHP
  - Python 2
  - Python 3
- Java solutions must be contained in a single class called `Solution` and must be run from the routine

  ```
  public static void main(String[] args)
  ```

  within this `Solution` class. If unsure, students should use the template solutions provided as a starting point for their solutions.

- Regarding the use of libraries or other external functions:
  - C and C++ programmers may only `#include` headers from the standard C and C++ libraries. In particular, C++ programmers are allowed to use the `string` class and container classes such as `vector` and `list`.
  - Pascal programmers may not import any units except for *Math*, *Strings* and/or *SysUtils*.
  - Java programmers may not use any classes aside from those in packages `java.lang`, `java.io` and `java.util`. Java programmers may not use dynamic loading of classes or any of the introspection features of the language. For instance, routines such as `Class.forName()` or classes such as `java.lang.ClassLoader` may not be used.
  - PHP programmers may not use any functions provided by extensions or external libraries.
- Programs must be single-threaded and single-process. For instance, C and C++ programmers may not call `fork()` or `system()`, and Java programmers may not use the class `java.lang.Thread` or call `Runtime.exec()`.
- Students may be disqualified if their programs:
  - attempt to read from or write to any files other than those specified in the problem statements
  - attempt to make network connections
  - contain any malicious code designed to harm or alter the judges' computer(s)
  - otherwise attempt to subvert the judging system.
- The source code for each solution must not exceed 100 000 bytes in size.

Time and Memory Limits

- Each program must run within the time and memory limits specified in the question statement. If, during judging, a program does not run within the time limit or uses more memory than permitted for a particular input file, it will receive a score of zero for that input file.
- Much of the judges' input data will be far more taxing than the sample input given in the question statements, and may push your program over the time limit. In this way, efficient programs will be rewarded.
- Judging will be performed on a 64-bit system with a clock speed no less than 2.0GHz, and all time limits refer to this judging machine.
- Programs written in Java, PHP or Python may run slower due to the overhead of the associated interpreters and/or virtual machines. The judges may at their discretion increase the time limits for these languages accordingly. *Contestants should note that this will not give these languages an advantage.*
- The memory limit is on the overall memory usage including executable code size, stack, heap, etc.

Judging

- Each question contains a number of subtasks, worth a total of 100 points. These subtasks and their point values will be described in the question. All questions are of equal value and all questions may be attempted.
- Programs will be compiled and run on the judges' machine(s):
  - GNU C/C++ Compiler 5.4.1. The C++11 standard will be used for C++ submissions.
  - Mono C# Compiler 3.2.8
  - Free Pascal Compiler 2.6.2
  - GNU Compiler for Java 5.4.1
  - PHP 5.5.9
  - Python 2.7.6
  - Python 3.4.3

  Students should specify the correct language for each submission using the dropdown box provided. Precise compiler flags used by the judges' machine(s) during judging can be found on the *Statement* tab of each problem.
- When judging a submitted program, each subtask is judged individually. For each subtask, several input scenarios will be presented to the program. The program successfully solves a subtask if it produces the correct output for each input scenario. The final score for the submission will be the sum of the point values of successfully solved subtasks. Students should submit a solution that solves as many subtasks as possible.
- A student's final score for each question will be the **maximum** among all of their submissions to that question.
- Please note that the scores shown to students during the contest are provisional only and are subject to change. The judges reserve the right to re-judge any submission or re-examine any student for any reason before declaring official results.
- In the event of ties, the judges reserve the right to either declare ties or to rank students using alternate means.

Why Did I Score Zero?

Candidates sometimes score zero even though they believe that they have a working solution to a problem. They are advised to check the *details* page of each submission to see the reasons why their solution was not judged as correct. An explanation of the messages on that page can be found on the *Documentation* tab. Below are some of the common reasons that good solutions score zero. Note that examples of solutions that score 100% in the various AIO languages can be found on the website `http://orac.amt.edu.au/aio/`.

Incorrect Input and Output Files

Each problem statement lists the names of its input and output files, similar to the example below.

Input File: *zeroin.txt*
Output File: *zeroout.txt*

In this example, if you try to open any of

- `"a:\zeroin.txt"`
- `"c:\mydir\zeroin.txt"`
- `"input.txt"`

then the file you are looking for will almost certainly *not* be on the judging machine and your program will score zero. Just open `"zeroin.txt"` without any additional directory information. The same goes for the output file.

Keyboard and Mouse Input

Your program should not be interactive. It should not have a graphical user interface. It should simply read from the input file, write to the output file, and exit. If your program requires any input from the user, the judging software will not supply this input and you will exceed the time limit. Examples include:

- 'Please enter the following value...'
- 'Press any key to exit...'
- Providing a form on which the user has to click a button to start the program.

Incorrect Output Format

Each problem is very precise about how the output file should be formatted. Your score is assigned by a judging program which tries to automatically extract your solution from your output file. Every problem statement includes sample input and output files, as a way of illustrating these formats. For Problem 2 in this paper, the first sample output file contains the single line:

```
4
```

In this example, the following output files would almost certainly score zero:

- `The answer is 4.`
- `"4"`

Compilation failed: Incorrect Java Class Name

When using Java, your code should be contained within a single class called `Solution`. If this is not the case, you are likely to receive a 'Compilation failed' verdict for your submission. Additionally, when you click on the *details* button, you will see the compiler output a message similar to `undefined reference to 'Solution::class$'`.

Incorrect Problem/Language Selected

You must submit your solution to the correct problem. Double-check that the name of the problem matches your solution. Additionally, you must select the correct language from the dropdown box when submitting your solution. This is especially important if you are using **Python** as you must specify if your solution is written in Python 2 or Python 3.

Subtasks

Points for submissions are awarded in *Subtasks*. Your solution must be judged as 'Correct' for **all** judges' testcases within a subtask for you to obtain those points. If your program exceeds the time limit, produces an incorrect output or a runtime error on even a single case, you will score zero for that subtask.

Violating Contest Rules

Each year a few programs are submitted that violate the contest rules. Be sure to read the *Program Restrictions* section of the rules, which details specific restrictions for each programming language. If you are unsure about anything before or after the contest then please email *aioquery@amt.edu.au* for clarification.