Australian Informatics Olympiad
Thursday 27 August, 2020

Information for Teachers and Students

Please read this information before the day of the contest

General Information

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.

Both students and teachers should familiarize themselves with the contest rules. Note that students may not see the actual contest problems until the beginning of the three-hour contest period.

Key links:

- AIO registration site: `http://aio.edu.au/register`
- Practice contest site: `http://aio.edu.au/practice`
- Contest site: `http://aio.edu.au/contest`

What's New in 2020

Here is a summary of the major changes being made in the 2020 AIO. This should not be viewed as exhaustive, and teachers and students should still read through all information carefully, even if they have competed or run the AIO many times previously.

- Students can sit the AIO from home. If they do, they will still need to be under supervision, by sharing their screen with their supervising teacher.
- Students should not be provided a physical copy of the paper. Instead, it should be downloaded from the contest website after the contest has begun.
- The versions of languages have been updated, and the compilers/interpreters used slightly changed.

Registration

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 make contact 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. See page 10 for Practice information. These login details will be used to log into the practice contest and also the actual 2020 AIO contest. **Students should not share these details with anyone**.

Eligibility

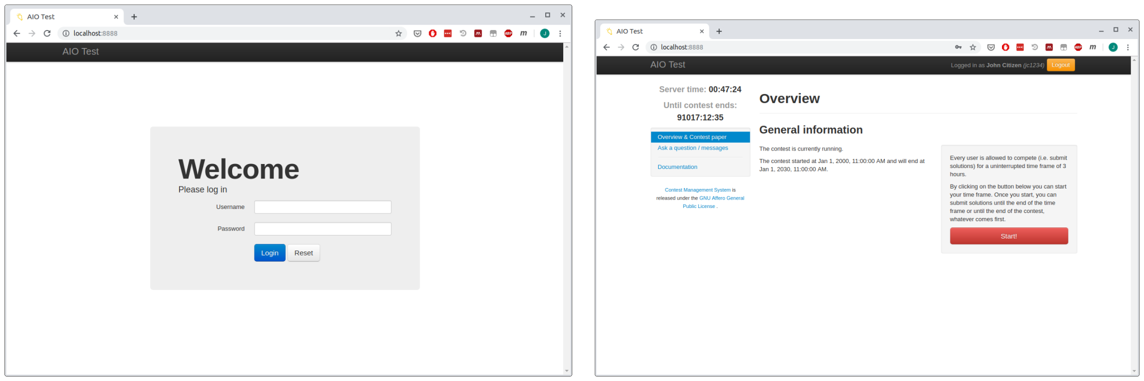
All students currently enrolled in an Australian secondary school (or equivalent overseas institution) are eligible to enter the 2020 Australian Informatics Olympiad.

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 5 pm inclusive, Australian Eastern Standard Time, on Thursday 27 August 2020. For instance, the contest may be held from 3 pm till 6 pm, but not from 5.30 pm till 8.30 pm.

Starting the Contest

- To start the contest, teachers should instruct students to log in to the contest system (at `http://aio.edu.au/contest`) with their username and password and click the red Start button to start their contest timer. At this point the students may download the contest papers and begin working.



The contest is currently running.

The contest started at Jan 1, 2000, 11:00:00 AM and will end at Jan 1, 2030, 11:00:00 AM.

Every user is allowed to compete (i.e. submit solutions) for a uninterrupted time frame of 3 hours.

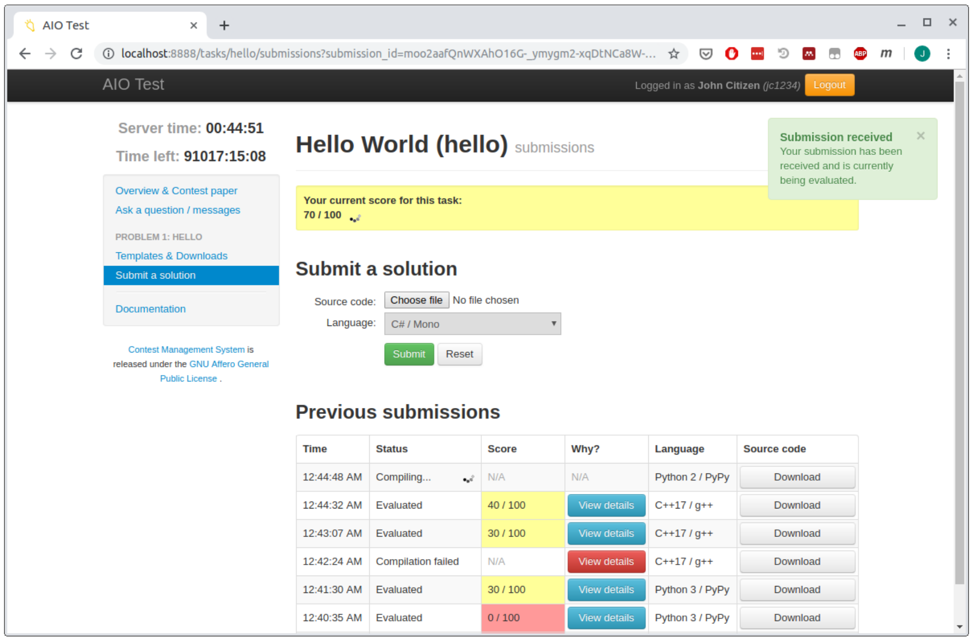
By clicking on the button below you can start your time frame. Once you start, you can submit solutions until the end of the time frame or until the end of the contest, whatever comes first.

Contest Management System is released under the GNU Affero General Public License .

Figure 1: Login screen and start button.

During the Contest

- Once the contest has started, students will be able to download an electronic copy of the contest paper, containing the statements for **all problems** from the *Overview & Contest Paper* page.
- Students will be able to download solution templates and sample data files for each problem, from the respective *Templates & Downloads* pages.
- Students should submit their solutions through this system **during the three-hour contest period**. See Figure 2.
- 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**.
- Students will be able to view the score of each of their submissions through the contest system. They can click the *View details* button to display the *Submission details* pop-up, breaking down the score for a submission. The pop-up also displays any output from the compiler (where applicable), **including warnings and/or compilation errors**. See Figure 3.



AIO Test

localhost:8888/tasks/hello/submissions?submission_id=moo2aafQnWXAhO16G-_ymygm2-xqDtNCa8W-...

AIO Test — Logged in as **John Citizen** *(jc1234)* Logout

Server time: **00:44:51**

Time left: **91017:15:08**

Overview & Contest paper

Ask a question / messages

PROBLEM 1: HELLO

Templates & Downloads

Submit a solution

Documentation

Contest Management System is released under the GNU Affero General Public License .

Hello World (hello) submissions

Submission received
Your submission has been received and is currently being evaluated.

Your current score for this task:
70 / 100

Submit a solution

Source code: Choose file No file chosen

Language: C# / Mono

Submit Reset

Previous submissions

Time	Status	Score	Why?	Language	Source code
12:44:48 AM	Compiling...	N/A	N/A	Python 2 / PyPy	Download
12:44:32 AM	Evaluated	40 / 100	View details	C++17 / g++	Download
12:43:07 AM	Evaluated	30 / 100	View details	C++17 / g++	Download
12:42:24 AM	Compilation failed	N/A	View details	C++17 / g++	Download
12:41:30 AM	Evaluated	30 / 100	View details	Python 3 / PyPy	Download
12:40:35 AM	Evaluated	0 / 100	View details	Python 3 / PyPy	Download

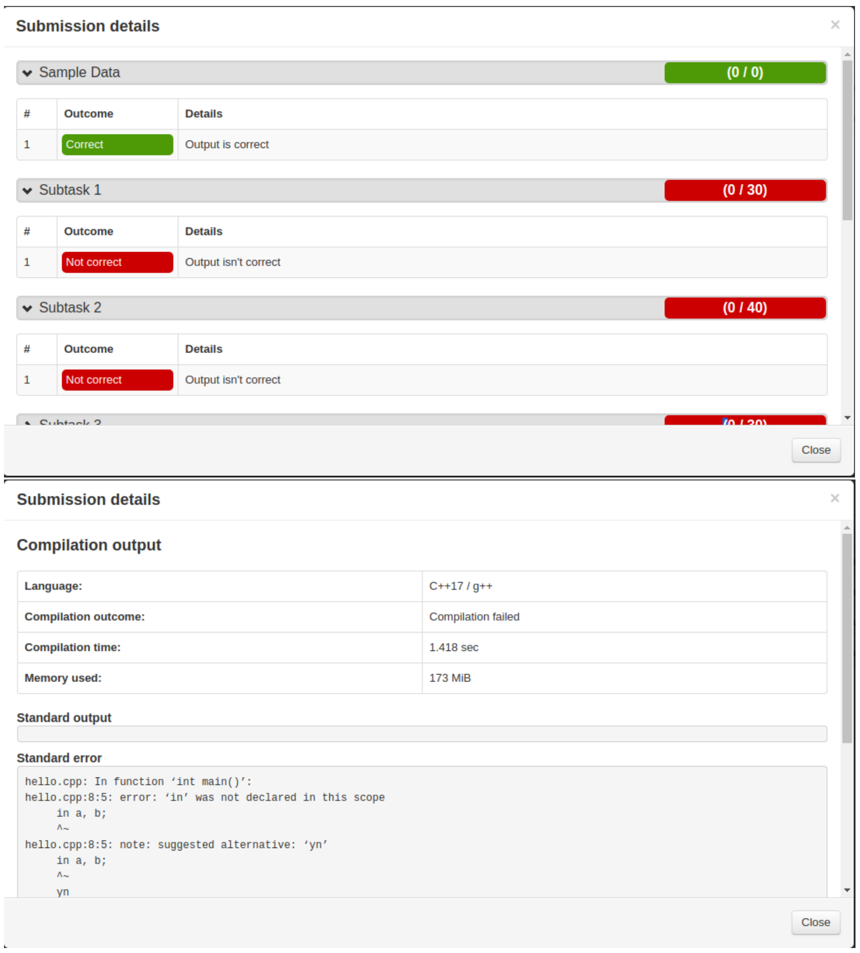
Figure 2: The *Submit a solution* page allows students to submit solutions to the corresponding problem by selecting a source code file and the language it was written in. It also displays the student's current score for each problem (yellow bar). See the Scoring section below for how final scores are determined.

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 28 August 2020.**
- Students should not discuss the problems in public forums/discussions, until Saturday 29 August 2020.
- Once results are finalised by the judges, students will be able to log back into the contest system to see their scores.

Queries and Difficulties

- If a teacher has 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 Mr Kevin Tran on 0421 386 818.



Submission details

Sample Data (0 / 0)

#	Outcome	Details
1	Correct	Output is correct

Subtask 1 (0 / 30)

#	Outcome	Details
1	Not correct	Output isn't correct

Subtask 2 (0 / 40)

#	Outcome	Details
1	Not correct	Output isn't correct

Submission details

Compilation output

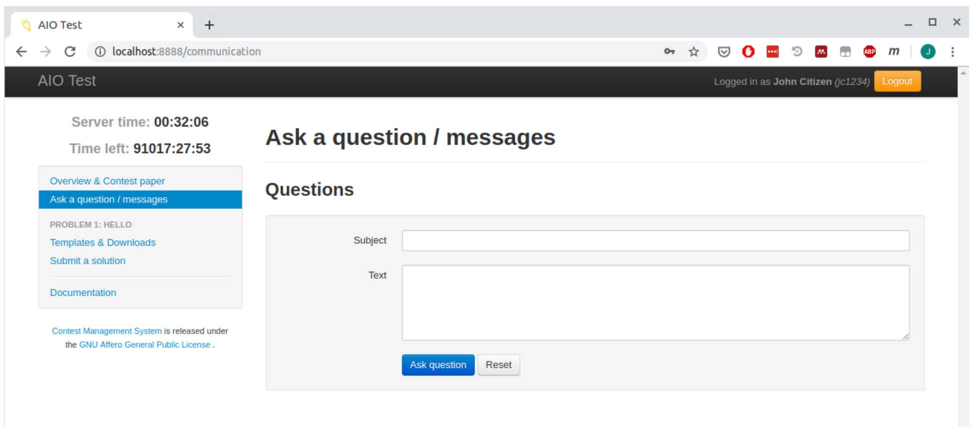
Language:	C++17 / g++
Compilation outcome:	Compilation failed
Compilation time:	1.418 sec
Memory used:	173 MiB

Standard output

Standard error

```
hello.cpp: In function ‘int main()’:
hello.cpp:8:5: error: ‘in’ was not declared in this scope
     in a, b;
     ^~
hello.cpp:8:5: note: suggested alternative: ‘yn’
     in a, b;
     ^~
     yn
```

Figure 3: *Submission details* pop-up, showing the score breakdown and compiler output of a student’s submission.



AIO Test
localhost:8888/communication
AIO Test
Logged in as John Citizen (jc1234) Logout

Server time: 00:32:06

Time left: 91017:27:53

Contest Management System is released under the GNU Affero General Public License .

Ask a question / messages

Questions

Figure 4: The *Ask a question/messages* page allows students to ask the judges questions.

Contest Rules and Environment

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 problems, and is fully responsible for clarifying or altering the rules or contest problems 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.

Supervision

Students must be in a supervised environment for the duration of the contest. If students are sitting the contest remotely, then they should share their screen with their supervising teacher.

Physical Materials

Each student should have access to one and only one computer. Additionally, students may bring:

- pens, spare paper, calculators and other stationery
- any number of books or other written material, including printed source code and handwritten notes

Students are forbidden from having electronic equipment other than their computer, and associated peripherals. Forbidden equipment includes mobile phones, tablets, and smartwatches.

Software and Electronic Materials

Students are free to use any text editors (e.g. Sublime, Notepad, TextEdit), IDEs (e.g. Code Blocks, Eclipse, XCode) or shells (e.g. WSL, MinGW, Terminal, Bash) they have installed on their machine. This includes any documentation, compilers, or templates that come installed with these.

Additionally, students may also access official language documentation on the following websites:

- C: `https://en.cppreference.com/w/c/language`
- C++: `https://en.cppreference.com/w/`
- C#: `https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/`
- Pascal: `https://www.freepascal.org/docs-html/ref/ref.html`
- Java: `https://docs.oracle.com/en/java/javase/11/docs/api/index.html`
- PHP: `https://www.php.net/manual/en/langref.php`
- Python 2: `https://docs.python.org/2.7/`
- Python 3: `https://docs.python.org/3/`

These sites will be linked to from the contest website, for ease of access.

Online text editors and IDEs (e.g. ideone, groklearning) are allowed, but please be very careful not to accidentally make any code you write public.

Students must not access any websites other than the competition website and the exemptions listed above. For example, students are forbidden from accessing:

- Q&A websites and forums (e.g. Stackoverflow, Reddit)
- source code sharing websites (e.g. GitHub, Pastebin)
- private messaging websites (e.g. Messenger, Instagram)

Communicating with other people

Students may ask their supervising teacher about administrative matters (e.g. *How much time do I have left? Can I go to the bathroom? I can't access the contest site?*).

Students may **not** ask other people about technical matters (e.g. *How do you program a* for *loop in Pascal? What does this paragraph in the problem mean? How many bytes does an integer take up?*).

Students must not communicate with other contestants by any means, including online messaging or email.

Clarifications and Announcements

During the contest, students can send a message using the *Ask a question/messages* page in the contest system, called a *clarification.* See Figure 4.

The judges may occasionally send public clarifications and announcements regarding the contest, which can also be found on this page.

- The judges will answer clarifications about the contest system, e.g. 'The system will not accept any more of my submissions, what's wrong?'
- The judges will also answer clarifications about the questions where they deem a question's statement to be ambiguous. These clarifications **must be phrased as a yes/no question**.
- The judges may, at their discretion, assist students whose technical issues or lack of technical understanding prevents them from attempting problems. The judges will not give hints towards how problems may be solved.
- The judges may ignore questions that are considered unrelated or solely argumentative.

Technical Issues

In the event of a contest website failure (e.g. the website goes down, incorrect submission results etc.) **students should continue to work on the problems on their local computers**. Although judging feedback is unavailable, students are still able to write code in their text editors or IDEs, or work on solving the problems on paper.

Once the technical issue has been resolved, extra time *may* be awarded to students who were adversely affected. To request extra time, the student should send a clarification detailing how they were affected by the technical failure, and the amount of extra time they are requesting.

In the event of a failure not related to the contest website (e.g. internet cuts out, computer crashes etc.), students should do their best to continue working on the contest. No extra time will be awarded for such failures.

Contest Format and Technical Details

There will be 6 problems in both the Intermediate and Senior divisions, to be completed in 3 hours. Each problem has a *statement* which describes a problem that you are required to write a program to solve. Students submit the source code for their program (called a *submission*) which is then compiled (if applicable) and run against a series of different input scenarios to test correctness and efficiency.

Students may make multiple submissions to a problem, of which the best submissions are used to calculate the total number of points the student will get for that question. Each problem contains a number of subtasks, worth a total of 100 points. These subtasks and their point values will be described in the problem. All problems are of equal value and all problems and subtasks may be attempted in any order. See the Scoring section for more details.

Program Restrictions

For each problem, students are required to write a program that solves the problem described. Each problem specifies:

- an input file to read the problem data from (e.g. `foodin.txt`)
- an output file to write the answer to (e.g. `foodout.txt`)
- a time limit that the program must run within
- a memory limit that the program must run within

Input/Output Format

The problem statement will describe the format of the input file your program must read from, and the desired output format of the output file your program should write the answer to. If you do not follow the expected format, you will likely score zero for that submission.

The contest system provides some leniency in the output format: it will ignore preceding and trailing whitespace (spaces & tabs) on each line.

Do not read from/write to any files other than the input and output files specified in the problem. In particular, any output to the screen will be ignored, and no input from the keyboard will be supplied.

Supported Languages

Your submission must be in one of the following languages:

- C11 (C language, 2011 standard)
- C++17
- C#
- Pascal
- Java 11
- PHP
- Python 2
- Python 3

When submitting a solution, students will need to specify which programming language they are using. Take care when choosing which version of Python (2 or 3) is used.

There are some additional instructions for Java:

- 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.
- 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.

If unsure, students should use the template solutions provided as a starting point for their solutions.

Compiler and Runtime Specifications

Programs will be compiled and run on the judges' machine(s) using the following software:

- C: GNU C Compiler 7.5.0, with flags `-DEVAL -std=gnu11 -O2 -pipe -static -s -lm`
- C++: GNU C Compiler 7.5.0, with flags `DEVAL -std=gnu++17 -O2 -pipe -static -s`
- Mono C# Compiler 4.6.2, with flag `-optimize+`
- Free Pascal Compiler 3.0.4, with flag `-dEVAL -O2 -XSs`
- OpenJDK / Javac 1.8.0_265
- PHP 7.2.24, with flag `-d memory_limit=-1`
- Python 2.7.17
- Python 3.6.9

Students may use any libraries, headers or packages that come standard with the compilers or runtimes specified above.

Forbidden Behaviours

- All 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()`.
- All programs must not attempt to make any network connections.
- All programs must not contain any malicious code designed to harm or alter the judges' computer(s), or otherwise attempt to subvert the judging system.

Submission

When submitting solutions, students should submit a single file (e.g. `file.cpp`) containing the source code that will solve the problem. In particular students should not submit the compiled executable (such as `file.exe`). **Compiled executables will result in a 'Compilation failed' verdict, and will receive a score of zero.**

- **You will be unable to make any submissions once the three hours are over!** Students should not leave all their submissions until the last few minutes—otherwise they risk running out of time. You should submit each solution once it is written. (You may always resubmit a better solution later.)
- Students may submit at most once per minute to each problem.
- The source code for each solution must not exceed 100 000 bytes in size.

Scoring

- A student's submission is compiled and run against several input scenarios to test the correctness and efficiency of the submission.
- These input scenarios are grouped into *subtasks*, which are each worth a proportion of the full 100 points.
- When judging a submission, 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 *all* input scenarios.
- For each input scenario, the program must run within the given time and memory limits specified in the problem statement. If, during judging, a program does not run within the time limit or uses more memory than permitted, it will fail that scenario and receive no points for the subtask.
- The final score for a problem will be the sum of the point values of all the subtasks that have been solved by *at least one submission.* This means students can write separate programs to solve different sets of subtasks, and their score will be automatically combined. The contest system will keep track of and display to the student their current score for a problem, at any point in time.
- 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.

Judging

- Much of the judges' input data will be far more taxing than the sample input given in the problem 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 Linux 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 is not guaranteed, and 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.

Practice 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.

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 *Submission details* pop-up 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* page. 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://aio.edu.au`.

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 a problem if, the 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 *View details* button, you will see the compiler output a message similar to `No class named Solution`.

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 drop-down 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.

Incorrect Mode for Output File

You must only open the output file for writing, not writing and updating. In particular, in Python you must open the output file with mode `w` not `w+`. This is one of the reasons you will receive 'Execution failed because the return code was nonzero' as feedback. The best way to avoid this type of error is to use the solution templates.

Subtasks

Points for submissions are awarded in *Subtasks*. Your solution must be judged as 'Correct' for **all** judges' test cases 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.

Recursion too deep in Python

When using Python, there is a limit on the depth of recursive calls, which defaults to 1000. If you are using recursion, you may wish to add the following lines to the start of your program, to increase this limit:

```
import sys
sys.setrecursionlimit(1000000000) # a large number
```

Note that if you recurse too deep (or infinitely deep), you may still consume time and memory exceeding the limits given in the problem statements before violating the increased limit. This will still result in your program crashing.

Violating Contest Rules

Each year a few programs are submitted that violate the contest rules. Be sure to read the Program Restrictions section, 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.