AUSTRALIAN MATHS TRUST | AUSTRALIAN INFORMATICS OLYMPIAD

AIO Rules and Requirements

Because of the technical nature of the competition, the AIO has specific rules and requirements that are additional to those of other AMT competitions. It is important that both teachers and students are aware of the requirements outlined in this document and the AIO Student Instruction Sheet.

Coordinators must also adhere to the general requirements of AMT competitions including ensuring students sit the comptiton in a controlled environment with face to face supervision. More information can be found in the competition handbook.

Physical materials

The AIO is run as an open book contest. Students may bring:

- pens, spare paper, calculators and other stationery
- handwritten notes
- textbooks
- written material including printed source code.

Each student should have access to only one computer.

Mobile phones, tablets and smartwatches are not allowed.

Electronic materials

Allowed

Students may use:

- text editors (eg. Sublime, Notepad or TextEdit)
- IDEs (eg. Code Blocks, Eclipse or XCode)
- shells (eg. WSL, MinGW, Terminal or Bash).

This includes any documentation, compilers, or pre-written code samples that come installed with the above tools. Students must take care not to accidentally make public any code they write using online text editors or IDEs.

Students may access the official websites of the competition's supported languages. Any pre-written code on these sites can be used during the competition. Links to these sites will be provided on the AMT student competition portal for easy access.

- C: https://en.cppreference.com/w/c/language
- C++: https://en.cppreference.com/w/
- Java: https://docs.oracle.com/en/java/javase/11/docs/api/index.html
- Python 3: https://docs.python.org/3/

Not allowed

Students must not use any pre-written code other than described above. This includes code pre-written by the student, a friend or teacher, or code from textbooks.

Students must not access websites other than the competition platform and the sites listed above. Students are forbidden from using any of the following tools during the competition:

- Q&A websites and forums (eg. Stack Overflow, Reddit)
- source code sharing websites (eg. GitHub, Pastebin)
- private messaging (eg. Messenger, Instagram)
- large language models and AI tools (eg. ChatGPT, GitHub, Copilot)

Program specifications

Each question includes a statement describing a problem that must be solved by submitting source code. Submissions are compiled by our system and run against a series of input scenarios to test correctness and efficiency.

Each problem statement specifies:

- a time limit for each test case
- a memory limit for each test case.

Problems are made up of subtasks. Problems and subtasks may be attempted in any order.

Students may make multiple submissions to a single problem or subtask. The best submission is used to calculate the number of points to be awarded.

Participants are provided with solution templates in each of the supported platforms. These are pre-written programs which can perform the necessary input and output. Students are **strongly encouraged** to use these templates as a basis for coding their own solutions.

Input and output format

The problem statement will describe the format of the input your program must read, and the desired output format your program should print. If you do not follow the expected format, you will likely score zero for that submission. The competition system allows some minor leniency in the output format. For example, it will ignore preceding and trailing spaces and tabs on each line.

Time limits

Each problem has a time limit per test case. For example, if the time limit is 1 second, then your program must take no more than 1 second to complete the test case. The time limit for a problem may vary depending on the language you are using.

The subtasks are designed to reward programs that can scale to larger inputs. A cleverer and more efficient algorithm will often score more points.

Submission

Students should submit their source code in a single file (eg. file.cpp, program.py)

Students should not submit compiled executable (exe) files. Compiled executables will result in a 'Compilation failed' verdict and receive a score of zero.

All programs must be single-threaded and single-process., 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 submit at most once per minute to each problem.

Programs must not:

- attempt to make any network connections.
- contain malicious code designed to harm or alter the judges' computer(s),
- otherwise attempt to subvert the judging system
- You cannot make submissions after the three hours have finished! Students should submit each solution once it is written and not leave all their submissions until the last few minutes. You can always resubmit a better solution later.

Java specific requirements

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 built-in packages except for java.lang, java.io and java.util.

Scoring

Problems are grouped into subtasks, which together make a total of 100 points.

Submissions are compiled and run against several input scenarios to test correctness and efficiency of each subtask. A subtask is solved correctly if the program produces the correct output for all scenarios.

The program must run within the time and memory limits specified in the problem statement. If not, the subtask will fail the scenario and receive no points.

The score for a submission is the sum of the point values of the subtasks that are solved by this submission. Students can make multiple submissions, and the score for the problem is the highest-scoring submission.

Usually the earlier, easier subtasks place more restrictions on the input, letting you make simplifying assumptions about the problem.

The student competition platform will display the score of each submission. Please note that the scores seen by students during the competition are provisional only and 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.

Scoring example

A problem asks you to find the median value in an array of N integers and includes the following description of subtasks:

For all test cases:

- 1 <= N <= 100 000
- The integers will be between 1 and 1 000 000 000 inclusive

Additionally:

- For subtask 1 (25 marks), N <= 1000, and N is always odd
- For subtask 2 (15 marks), N <= 1000
- For subtask 3 (20 marks), N is always odd
- For subtask 4 (40 marks), no special constraints apply

If you coded a program which worked correctly and within the time limit for all test cases where N <= 1000, you could expect to solve subtasks 1 and 2, for 25 + 15 = 40 marks.

If you coded a program which worked correctly and within the time limit for all test cases where N is odd, you could expect to solve subtasks 1 and 3, for 25 + 20 = 45 marks.

If you submitted both programs above, you would earn 45 marks for the problem, because the highest-scoring submission earned 45 marks.

Compiler

Programs are compiled and run 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
- OpenJDK / Javac 11.0.11
- Python 3.9.19 / PyPy 7.3.16
- Students may use any libraries, headers or packages that come standard with the compilers or runtimes specified above.

Judging

Students are only marked on the correctness and speed of their programs. However good coding style may be advantageous as students may find it easier to debug and it may be less likely to be pushed over the time limit by our judging system.

Judging will be performed on a 64-bit Linux system with a clock speed no less than 2.0GHz. All time limits refer to this judging machine.

Programs written in Java or Python may run slower due to the overhead of the associated interpreters and/or virtual machines. The judges may increase the time limits for these languages at their discretion. This is not guaranteed and will not give these languages any advantage.

The memory limit is an overall memory usage including executable code size, stack, heap, etc.

Email: competitions@amt.edu.au
Phone: (02) 6201 5136
Administration portal: https://competitions.amt.edu.au/admin/
Student adminstration portal: https://competitions.amt.edu.au/

Postal: AIO Competitions Team
Australian Maths Trust
170 Haydon Drive
Bruce ACT 2617