The 3rd Universal Cup Finals

Technical Notes (v1.0)

The Universal Cup Technical Committee

April 19th, 2026

This document contains important technical information related to the Universal Cup Finals programming environment. It is important that your team read and understand all the information below.

1 Programs

- The languages allowed in the contest are C, C++, D, Java, Kotlin, Python 3 and Rust.
- There is a limit of 256 kibibytes on the length of file submitted for judging.
- Your program must read its input from "standard input".
- Your program should send its output to "standard output". Your program may also send output to "standard error", but only output sent to "standard output" will be considered during judging. (Note that sending too much output to "standard error" might be harmful, in the sense that it can slow your program down.)
- If your program exits with a non-zero exit code, it will be judged as a **RUN-ERROR**. This can have a lot of different causes like division by zero, incorrectly addressing memory (e.g. by indexing arrays out of bounds), trying to use more memory than the limit, reading or writing to files, etc.
- Programs submitted to the judges will be run inside a "sandbox".
  - The sandbox will allocate memory for your program as specified in each problem.
  - Your entire program, including its runtime environment, must execute within the specified memory limit for the problem. For interpreted languages (Java, Kotlin and Python), the "runtime environment" **includes** the interpreter (that is, the JVM for Java and Kotlin and the Python interpreter for Python).
  - The sandbox memory allocation limit will be the same for every language.

- The command and command-line arguments used to invoke your program within the sandbox are the same as those given below.
- Programs running in the sandbox will be "pinned" to a *single* CPU.

2 IDEs and Editors

- The following IDEs (Integrated Development Environments) are preinstalled on the contest system: **CLion**, **Code::Blocks**, **Eclipse**, **IntelliJ IDEA**, **PyCharm**, **RustRover**, **VS Code**.
- The following editors are preinstalled on the contest system: **Vim**, **Gvim**, **NeoVim**, **Emacs**, **Text Editor (GEdit)**, **Geany**, **Kate**.
- You are free to install plugins for the editor or any other software you like from the Internet. See the next section for instructions on using root privileges during installation.

3 Compiler

The detailed compiler versions are listed below. You can also find them at **Language** page in DOMjudge.

C	`gcc-14 (Ubuntu 14.2.0-4ubuntu2~24.04.1) 14.2.0`
C++	`g++-14 (Ubuntu 14.2.0-4ubuntu2~24.04.1) 14.2.0`
D	`DMD64 D Compiler v2.112.0`
Java	`openjdk 21.0.10 2026-01-20`
Kotlin	`kotlinc-jvm 2.3.20 (JRE 21.0.10+7-Ubuntu-124.04)`
Python	`Python 3.11.15 (7.3.21+dfsg-4~ppa1~ubuntu24.04, Mar 24 2026, 00:09:28)`
Rust	`rustc 1.95.0 (59807616e 2026-04-14)`

You are free to execute (test) your programs on your machine using any method you choose. However, it is recommended that you compile and execute your programs using the command line described below since this will ensure the closest match to the way in which the judges will compile and execute your programs.

Language		Command[1]
C		`gcc-14 -x c -O2 -std=gnu23 -static ${files} -lm`
C++		`g++-14 -x c++ -O2 -std=gnu++23 -static ${files}`
D		`dmd -O -release -inline -boundscheck=off ${files}`
Java	Compile	`javac -encoding UTF-8 -sourcepath . -d . ${files}`
	Execute	`java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m ${mainclass}`
Kotlin	Compile	`kotlinc -d . ${files}`
	Execute	`kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m ${mainclass}`
Python[2]	Compile[3]	`pypy3 -m py_compile ${files}`
	Execute	`pypy3 ${mainsource}`
Rust		`rustc --edition=2021 -C opt-level=3 ${files}`

4 Root Privileges

- You may use `sudo` to obtain root privileges so that you are free to install other software you like from the Internet. The password will be given during the warmup session.
- You should not use root privileges to make malicious changes to your workstation (including but not limited to affecting other teams and making staff lose control of your workstation).
- You are responsible for any consequences of improper use of root privileges. We recommend that you use root privileges only during the workstation configuration stage.
- Any behavior that is deemed an activity detrimental to the contest may result in disqualification, determined by the staff.

5 Submission

- Programs are submitted to the judges using the DOMjudge contest control system.
- To access DOMjudge, use the *Contest* > *DOMjudge* menu item or desktop entry.

[1]The notation "`${files}`" means "the list of file names passed to the corresponding script as arguments". For Java and Kotlin submissions, the notation "`${mainclass}`" refers to the name of the main class in your program. For Python submission, the notation "`${mainsource}`" refers to the file name of the entry point in your program.

[2]Please refer to the appendix for the list of installed `pypy3` modules.

[3]Python programs will be "syntax checked" when submitted; programs which fail the syntax check will receive a "**COMPILER-ERROR**" response (for which no penalty applies, just as with other programs which fail to compile).

- See the separate *DOMjudge Team Guide* for details on using DOMjudge.

6 Printing

- There will be runners who will deliver printed output to your team workstation (teams will not have direct access to the printers).
- You can upload your source code file in DOMjudge **Print** page.
- To print from the command line, type `printfile --help` for detail.
- Only source code (plain text) can be printed. Please **DO NOT** submit non-text files (e.g. ZIP, PDF, etc.).
- Printing files from within IDEs and other applications via system dialog is **NOT supported**.
- Print jobs are limited to a few pages long; printing excessively long output will be deemed an activity detrimental to the contest and subject to disqualification.

7 Sample Data and Problem Statements

- The time and memory limits and the sample data are available in DOMjudge **Problemset** page.