icpc.foundation

ICPC North America Regionals 2020

icpc international collegiate programming contest

icpc global programming tools sponsor

ICPC Pacific Northwest Regional Contest

TECHNICAL NOTES

This document contains important information related to the Pacific Northwest Regional Contest environment. It is important that your team read and understand all the information below.

All Programs:

- The languages allowed in the contest are C, C++, Java, Kotlin, and Python 3.
- There is a limit of 65,535 bytes on the length of files 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.
- If your program exits with a non-zero exit code, it will be judged as a Run Time Error.

C/C++ Programs:

- Use the filename extension “**.cpp**” for C++ program files (extensions **.cc**, **.cxx**, and **.c++** can also be used). Use the filename extension “**.c**” for C program files.

Java Programs:

- Do not use *package* statements (that is, your solution must reside in the “default package”.) Use the filename extension “**.java**” for all Java source files.

Kotlin Programs:

- Do not use *package* statements (that is, your solution must reside in the “default package”). Use the filename extension “**.kt**” for all Kotlin source files.

Python Programs:

- In conformance with World Finals rules, only Python 3 (but not Python 2) is supported. Use the filename extension “**.py**” for all Python source files.
- Python programs will be “syntax checked” when submitted; programs which fail the syntax check will receive a “Compilation Error” response (for which no penalty applies, just as with C/C++/Java/Kotlin programs which fail to compile). See the sections below for information on how to perform a syntax check yourself in the same way as will be done by the Judges.

Command-line Usage for C/C++:

- To compile a C or C++ program from a command line, type the command

```
compilegcc progname.c
```

(for C programs) or

```
compileg++ progname.cpp
```

(for C++ programs)

where **`progname.c`** or **`progname.cpp`** is the name of your source code file.

The **`compilegcc`** command is a script which invokes the GNU GCC compiler with these options:

```
-x c -g -O2 -std=gnu11 -static ${files} -lm
```

The **`compileg++`** command is a script which invokes the GNU G++ compiler with the same options as those used by the Judges:

```
-x c++ -g -O2 -std=gnu++17 -static ${files}
```

- To execute a C/C++ program after compiling it as above, type the command

```
./a.out
```

Command-line Usage for Java:

- To compile a Java program from a command line, type the command

```
compilejava Progname.java
```

where **`Progname.java`** is the name of your source code file. This will compile the source code in the file **`Progname.java`**, and will produce a class file named **`Progname.class`**. The **`compilejava`** command is a script which invokes the **`javac`** compiler with these options:

```
-encoding UTF-8 -sourcepath . -d . ${files}
```

- To execute a Java program after compiling it, type the command

```
runjava Progname
```

where **`Progname`** is the name of the class containing your **`main`** method (your source code file name without the filename extension). The **`runjava`** command is a script which invokes the **`java`** command with the following options:

```
-XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m ${mainclass}
```

Command-line Usage for Python 3:

- To “compile” (syntax-check) a Python 3 program from a command line, type the command

```
compilepython  progname.py
```

where **`progname.py`** is the name of your Python 3 source code file. The **`compilepython3`** command is a script which invokes the **`pypy3`** Python 3 interpreter as follows:

```
pypy3 -m py_compile ${files}
```

which compiles (but does not execute) the specified Python program and displays the result (i.e., whether the compile/syntax-check was successful or not).

- To execute a Python 3 program from a command line, type the command

  ```
  runpython  progname.py
  ```

  where `progname.py` is the name of your Python 3 source code file. The `runpython3` command is a script which invokes the `pypy3` Python 3 interpreter passing to it the specified Python program file.

Command-line Usage for Kotlin:

- To compile a Kotlin program from a command line, type the command

  ```
  compilekotlin  progname.kt
  ```

  where `progname.kt` is the name of your Kotlin source code file. The `compilekotlin` command is a script which invokes the `kotlinc` compiler with these arguments:

  ```
  -include-runtime -d . ${files}
  ```

- When compiling Kotlin either with `compilekotlin` or by hand, you may get warnings about an illegal reflective access operation. These warnings can be ignored.

- To execute a Kotlin program from a command line, type the command

  ```
  runkotlin progname.jar
  ```

  where `progname.jar` is the name of the jar file created by `compilekotlin`. The `runkotlin` command is a script which invokes the Java JVM with the following options:

  ```
  -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m ${mainclass}
  ```

IDEs and Editors

- The following IDEs (Integrated Development Environments) are available on the contest system: **CLion**, **Code::Blocks**, **Eclipse**, **IntelliJ IDEA**, **PyCharm, VS Code**. They can be accessed using the *Applications□Programming* menu.

- The following editors are available on the contest system: **Vim**, **Gvim**, **Emacs(GUI)**, **Emacs(Terminal)**, **GEdit**, **Kate**. They can be accessed using the *Applications□Accessories* menu.

Documentation

- Documentation for each available programming language can be found on your machine under the *Applications□Programming□Documents* menu.

- Additional documentation, such as these Technical Notes, is also available under the *Applications□Programming□Documents* menu.

Submissions

- Programs are submitted to the Judges using the *PC^2 WTI* contest control system. PC^2 can be access using the *link provided at the start of the contest*. For details on using PC^2, see the separate *PC^2 WTI Team Guide*.

Scoreboard

- The current contest scoreboard (standings) can be viewed using the *Contest□Scoreboard* menu. Note that *the scoreboard will be frozen with one hour remaining in the contest.*.

Sample data and Problem Statements

- The problem statements and sample data given in each problem will be accessible in machine-readable form via a link on the Scoreboard. To access the sample data, go to the *Contest□Scoreboard* menu, follow the Scoreboard link to your Division, and click on the “Sample Data” link. This will download a .zip file containing the sample data for all problems in that Division. Unzip that file under your home directory to obtain the sample data.