Computer Programming Olympiad

A project of the Institute of IT Professionals South Africa

Ph: 021-448 7864 • Fax: 021-447 8410 • PO Box 13013, MOWBRAY, 7705 • info@olympiad.org.za • www.olympiad.org.za

PROGRAMMING OLYMPIAD
FINAL ROUND 2015

Environment Manual

Contents

1. Introduction

This document describes the environment you will be using for the Final Round of the South African Programming Olympiad. Section 2 describes the machines you will be using, where to save files etc, without dealing with programming. Section 3 describes the compilers and development environments. You will also be using a web-based hand-in system, described in section 4. Specific issues related to the interaction between your programming environment and the web server may be found in section 5.

2. Local environment

You will be using an Ubuntu Linux workstation attached to the UCT network. Before each competition round the contest organisers will log you into a workstation. If technical difficulties occur, you may have to move to another workstation. Because of this, you should not store any important files anywhere except your home directory. This directory is called */home/username* and is stored on a central fileserver. Here *username* would be the *username* that you were logged on with. You may get the path of your current directory by running 'pwd' in a terminal. **You should save all your work often to the home directory.**

You should not logout from your workstation or reboot your computer during the contest. If you experience technical problems with your computer, ask an organiser for assistance. You will usually be moved to another computer.

There will be shortcuts to various tools that will be useful during the contest in the Applications Menu and on the Desktop.
There will also be a shortcut set up on the Desktop for the contest webpage.

3. Compilers and IDEs

3.1 C++

For C++, you will be using the C++11 standard as implemented by GCC 4.8. This version of GCC adheres particularly strictly to the C++ specification. Some things that may work in other compilers that will not work under GCC include:

- main *must* return int, and *must* return 0. This might not be enforced by GCC, but failure to do so can lead the marking system to think that your program crashed (in which case you score 0).

• stricmp does not exist; use strcasecmp instead

• strrev does not exist; write one yourself

Note that using the `endl` operator in an I/O stream causes the stream to be flushed, which has a performance overhead. It is faster to use `"\n"` instead.

3.2 Java

You will be using the Open JDK 7. You should be aware that while the Scanner class makes I/O easier, it is roughly 10 times slower than using a BufferedReader together with a StringTokenizer. **For tasks with large amounts of input, it may not be possible to obtain a full score using Scanner.**

3.3 Python

Python version 3.4 will be used both in the competition environment and the Linux-based hand-in system. The supported IDE is IDLE.

3.4 IDEs and Text Editors

The competition workstations include the following editors:

- Gedit
- Vim 7.4
- Emacs 24.3
- Kate 3.13
- Geany 1.23
- Eclipse 3.8
- Netbeans 7.4
- IDLE for Python 3

You are allowed to use any of these to write your code. The organisers will attempt to help with any technical difficulties you may encounter while using your choice of editor or IDE. However, if a technical problem with your IDE causes you to lose time during the contest, you will not be provided with extra time. You may also be expected to use a different IDE or editor if the problem cannot be resolved.

For Java developers who are unfamiliar with Eclipse:
When Eclipse starts, you will be presented with a welcome screen, which you can close. Select File ! New ! Project, then Java project. For the project name, use the short name of the problem. Once you have created a project, you can add a file by selecting
File > New
> Class.

4. Web-based hand-in system

The South African Programming Olympiad uses a web-based hand-in system, similar to the one used at the International Olympiad in Informatics. A version of the source code can be found at <https://github.com/cms-dev/cms>. The web server can be accessed via the link found on the desktop. The first time you connect, you will be warned that the certificate is self-signed; select to accept the certificate permanently and continue.

You will be allocated a username (initial and then surname - if your name is Raymond Luxury-Yacht, it will be rluxuryyacht) and a password for accessing this system. Once you have logged in, there are a number of menu options available on the left:

Overview This provides key details about the contest. Most notably, you will be able to see time and memory limits (without language modifiers) here. You will also be able to see the start and end times of the contest.

Communication Do not use this page. If you need to submit a question, please use the paper provided by the organisers.

Statement. For each problem, you will be able to download task statements from the statements page.

Submissions This page allows you to hand in solutions, as well as to view files that you have already handed in.

When you hand in source code as a solution, your source code will be compiled and run on all test cases. You will be able to see a summary of your program's score, including how it did on each test case. Your final score for a problem will be the maximum score across all your submissions. Please be advised that the server may take a while to run your program on every test case. Please be prepared to wait up to 10 minutes before seeing your score.

You will be allowed to submit at most 100 times (across all problems) during the contest. You will not be allowed to submit more than once every 60 seconds. The organisers reserve the right to change these limits during the contest to ensure that the submission system will give quick feedback to all participants.

Testing You can use this page to see how your program will run on the server, on test data other than the sample test case.

You will be allowed to test at most 100 times (across all problems) during the contest. You will not be allowed to test more than once every 60 seconds. The organisers reserve the right to change these limits during the contest to ensure that the submission system will give quick feedback to all participants.

5. Miscellaneous

The hand-in system is running on Ubuntu 14.04 LTS. Because of the difference in system setup, bugs in your program may manifest themselves in different ways on the hand-in system, perhaps not even appearing at all on the local system. Here are some possible reasons for these differences in behaviour.

Memory limitations When run on the server you are limited to a fixed amount of memory (usually 256MiB). If you try to use more than this, then

C++ new will throw a bad_alloc exception
Java new will throw java.lang.OutOfMemoryError
Python Operations will throw MemoryError or just crash Python

If you use up all the space with global variables in C/C++, your program will probably be killed before any of your code executes.

Note that there is some overhead for your code, stack and so on, so not all of the memory is available to you for variables. Count on losing at least 2–5MB to overheads.

You will usually be given far more memory than you actually need, so overheads should not be a major issue.

Python is the worst culprit for memory consumption. Lists are very memory-inefficient and will consume several times as much memory as one would expect.

Memory accesses (C++) If you have been used to programming C++ on Windows then you may find that Windows tends to be a lot more forgiving than Linux when it comes to memory accesses. In particular it will generally let you read from an invalid address. Sources of invalid addresses include NULL pointers, uninitialised pointers, pointers to memory that have been freed, and out of range array indices. If your program is crashing with signal SIGSEGV (known as a segmentation fault) on the server then this is a likely cause of the fault.

Uninitialised memory (C++) If you use a variable without initialising it, your program might behave differently depending on what is in that variable when it starts. You can often get the compiler to warn you of this by specifying the compiler flags -O2 -Wall (on GCC), although it is not perfect.

Compiler flags Read the rules to determine the compiler flags that are used on the server. You may find that the only reason a bug appears on the server and not your machine is that you were not using the same compiler flags. This is usually a sign that you have done something wrong, such as accessing memory out of bounds (where the flags might change the way memory is laid out) or not initialising a variable (without the optimisation flags the compilers may zero the memory for you, but will skip this when optimising).

Exit code A program that returns with a non-zero exit code is considered to have crashed.
C++ programmers must declare their main function to return int and must explicitly return 0 at the end of the main function.

Headers, units and modules Some C++ headers and Python modules are specific to Windows. You may find that the server will complain about a missing header file - if so then this is probably the problem. A good example of this is the <windows.h> C header.

Some Java textbooks use the add-on package java.lancs to provide simple input routines. This package is not part of the Java standard and will not be available on the server. You should instead use BufferedReader or Scanner. We can assist you with this at the practice round, but no assistance will be given during the contest.

5.1 Documentation

Documentation for all the languages may be found on the hand-in server using the link provided on the Desktop. This includes documentation on the C++ Standard Template Library.

6. Summary

Hand-in server account initial and surname

Shortcuts Desktop and Menu

Home directory /home/username