Lab #0: Trial Lab

CS1010 AY2013/4 Semester 1

Date of release: 16 August 2013, Friday, 10:00hr.

Submission deadline: Week 3, during your discussion session.

School of Computing, National University of Singapore

0 Introduction

This is a non-graded lab. However, you need to submit your program on **CodeCrunch** in the presence of your discussion leader (DL) during your first discussion session, to show that you know how to do it.

This lab requires you to do only 1 exercise.

If you have any questions on any lab exercise, please may post your queries **on the relevant IVLE discussion forum**. Important: Do **not** post your programs (partial or complete) in the forum before the deadline!

---

1 Exercise 1: Volume of a Box

1.1 Learning objectives

- Using CodeCrunch.
- Run through edit-compile-run cycle of program development.
- Using the UNIX environment.
- Detection and correction of errors in a program.

1.2 Task statement

Write a program **`box_volume.c`** that reads three positive integers representing the length, width and height of a box, and computes the volume of the box.

You may assume that the volume of the box does not exceed the maximum value representable in the `int` data type. (What is that value? See 1.5 Exploration below.)

Check sample run for input and output format, and submit your program through CodeCrunch.

1.3 Sample run

Sample run using interactive input (user's input shown in blue; output shown in **bold purple**). Note that the first two lines (in green) below) are commands issued to compile and run your program on UNIX.

```
$ gcc -Wall box_volume.c -o box_volume
$ box_volume
Enter length: 12
Enter width : 3
Enter height: 10
Volume = 360
```

1.4 Important notes

- You may download a skeleton program `box_volume.c` from the next section (section 1.5). The skeleton file is almost complete and you just need to correct a few statements in the program.

- CodeCrunch awards marks for correctness ONLY if your output adheres to the given format. Hence, do not add any other characters (such as blanks that are not asked for in your output, or change the spelling in your output. The following outputs will all be graded as **incorrect** for this exercise:

    - `volume = 360` (reason: "Volume" mis-spelt as "volume")
    - `Volume=360` (reason: spaces around = sign missing)
    - `   Volume = 360` (reason: additional spaces before "Volume")
    - `Volume   =   360` (reason: too many spaces around = sign)
    - `Volume = 360.` (reason: additional dot at end of line)

- The last output statement in your program must end with a new line character '\n'. Otherwise, it is possible that your program may fail the correctness test of CodeCrunch.

1.5 Skeleton file and sample test data

- box_volume.c
- Input files: box1.in | box2.in | box3.in | box4.in | box5.in
- Output files: box1.out | box2.out | box3.out | box4.out | box5.out

1.6 Explanation on input and output files

Note that in Section 1.5 above we provide the input and output files for your ease of checking. In writing your programs, you may assume that the input data are entered interactively (through stdin, i.e. the keyboard), and the outputs of your program are displayed on the monitor (stdout).

CodeCrunch takes your submitted program, compiles and runs it on each of the test data input files provided, using a technique called input file redirection.

You can use the technique too. If you have an input file, say, **box1.in**, that contains the input data, you could make your program reads in data from this file instead of entering the data interactively through the keyboard. For example, assuming that your program is called **a.out**, the following command:

```
a.out < box1.in
```

runs **a.out** which draws the input data from the file `box1.in`.

You may also use output redirection to redirect the output of your program to a file, instead of showing it on the monitor. For example:

```
a.out < box1.in > 1.out
```

The above command saves the output of your program to the file `1.out`. You may then compare the content of this file with the content of the provided output file `box1.out` with the expected answer, by using the UNIX `diff` command as follows:

```
diff 1.out box1.out
```

If the `diff` command does not produce any output, it means that the two files `1.out` and `box1.out` are identical.

1.7 Number of submissions

For this exercise, the number of submissions is **99**.

1.8 Exploration

What is the largest value available for `int` data type? Search the Internet for the answer, and write a separate program (no need to submit this) to test it out.

---

2 Deadline

You must submit the program on CodeCrunch in the presence of your DL during your first discussion session.

---

3 Reading

Read up Lab Guidelines to prepare yourself for subsequent lab assignments.

- 0 Introduction
- 1 Exercise 1: Volume of a Box
  - 1.1 Learning objectives
  - 1.2 Task statement
  - 1.3 Sample run
  - 1.4 Important notes
  - 1.5 Skeleton file and sample test data
  - 1.6 Explanation on input and output files
  - 1.7 Number of submissions
  - 1.8 Exploration
- 2 Deadline
- 3 Reading

Aaron Tan
Monday, July 22, 2013 04:10:37 PM SGT