

C Programming: *Self Test*

1 Introduction

Since the Embedded Systems Laboratory (ESL) is project based, centered around writing the low-level control software for stabilizing a flying quadcopter, it is imperative that you master the C programming language to a decent extent. If you have completed the CSE2425 Embedded Software course, you automatically qualify for the ESL. If not you will have to demonstrate your coding skills by means of an online test. Basically you have to write a C program that counts words on the standard input, and submit that electronically for testing. Upon failure to pass our benchmark tests, you may retry. However, as the feedback may be minimal (“you failed”), it is advisable to follow best practices and write your own test suite.

If it takes you more than 10 retries to pass the test you are advised to take the (full) CSE2425 Embedded Software course in Q2, which may be part of your ES homologation, or at least the first 3 weeks of it (covering C programming and associated tools).

2 Counting Words

Implement a program that counts the number of times each word appears in the standard input. A word is defined as a sequence of one or more alphanumeric characters. You may use `scanf`¹ to read a particular set of characters from the input, but other solutions are also accepted (All ways lead to Rome). Storing the words should be linear in the size of the input, so simplistic approaches like statically allocating big tables will not be approved. Instead `malloc` and `free`² must be used to implement a dynamic data structure like a tree or hashmap storing the words and associated counts. Given the input:

```
foo bar_, foo!  
bar  "baz".  
foo?
```

The program should write the following to the standard output:

```
bar: 2  
baz: 1  
foo: 3
```

The words must be printed in alphabetical order, one per line.

¹<http://en.cppreference.com/w/c/io/fscanf>

²<http://en.cppreference.com/w/c/memory>

3 Submission

Assignments should be uploaded to CPM³; please follow these instructions:

- Enroll yourself for the 'CS4140ES C-programming Test' in CPM.
- Put all the source code inside a file called `wordcount.c`;
- Upload this `wordcount.c` file to CPM.

Your code will be tested automatically by CPM, and a report of its success or failure will be created. You may want to enable notifications in your CPM preferences such that you will be sent an email once the result is available (submissions are graded in batch mode for technical reasons). If your program fails to produce the right output on a test case, you will be provided with a diff⁴ listing showing the differences between your output (lines marked with a `<`) and the correct output (lines marked with a `>`).

Once your program passes the automatic test (status 'ScriptApproved') you can register yourself with the CS4140ES course (in due time; it runs in Q4).

³<http://cpm.ewi.tudelft.nl>

⁴https://en.wikipedia.org/wiki/Diff_utility