

**31251 – DATA STRUCTURES AND ALGORITHMS
32510 – PRINCIPLES OF OO PROGRAMMING IN C++
AUTUMN 2009**

ASSIGNMENT 3 - DUE 11:59pm, 3/6/2011

This assignment is worth 25% of the total marks for this subject.

Due Date

- The final due date for this assignment is 11.59pm, 3/6/2011.
- However, a preliminary draft of the program is due one week earlier by 11.59pm 27/5/2011. Failure to have the preliminary draft submitted in time will result in marks lost. See the assessment section below for what defines a preliminary draft and the marks that will be lost.

Requirement

For this assignment, you are to write a program that reads a text file, identifies the words in it and then checks the spelling of each word against a small dictionary of words. Any words not found the dictionary are printed out. Finally, the total number of words in the file is printed out and the total number with unknown spelling.

Definitions

Text Files:

For our purposes, a text file consists of lines of text containing words, spaces and punctuation, and with a newline character on the end of each line. All characters will be from the basic ASCII character set.

Words:

- Words are sequences of alphabetic and numeric characters, the single quote, the underscore and hyphen characters that are separated by sequences of one or more separation characters. See below for a list of the separator characters.
- The input for this assignment will consist of normal words plus integer and floating point numbers.

Implementation Details

Data Structure – Template Binary Tree

Your program must use a template binary tree, with each node in the list containing a C++ *string* class. Do not use other lists such as doubly linked lists, etc. A copy of the `bintree.h` and `binnode.h` files can be found in the `/pub/prprog/assign/assign3` directory and the DSA web site in the `asassign3` directory.

Please note: You must use these specific classes from these directories classes and you may not alter them in any way.

Program Usage

Assuming your compiled program is called `a.out` your program will use command line argument as follows

```
./a.out textfile
```

where `textfile` is the file containing the text you are going to spell check.

Program Operation

All the code for your program will be contained within one file. This is a limitation imposed by the submission system. Do not copy and paste the `bintree` and `binnode` code into your file. Just `#include` it.

In the `/pub/prprog/assign/assign3` directory or the DSA `assign3` directory you will find a file called `dict.txt`. This file contains the 1000 most commonly used words in the English language with one word per line. Place this in the directory where your program resides. Do not change the contents of it.

The program should start by reading the dictionary file, loading each word into a C++ string and then storing the string in a binary tree.

Your program will then read the text file, line by line, and extract every word from the line. Each extracted word will be checked to see if it exists in the tree. If not then the word will be printed. Please note the following regarding extracted words

- Before checking a word against the tree, all letters in it will be changed to lower case. That is “Hello” will be changed to “hello” before seeing if it exists in the tree.
- If any extracted word is a number, either an integer or real, it will be ignored. That is, it won’t be checked for spelling. Look up the `strtod` library function to work out how to do this.

Separator Characters

- The list of word separator characters should include: space, tab, newline, carriage return and the common punctuation characters
`.,:”~!#%^*()=+[]{} \<>?/`
- If the `.` character has a space, tab, newline or digit on the left and a digit on the right then it is treated as a decimal point and thus part of a number. Otherwise it is treated as a full stop and a word separator.
- The tab character is represented by the escape sequence `\t`.
- The carriage return is used in PC environments. It is represented by the escape sequence `\r`.

Benchmark Program

To clarify how the program works, a benchmark executable version has been placed on the server. You can run it by typing the following command

```
/home/glingard/spellcheck
```

To give you an idea of the size of this assignment, the source code for the benchmark program, including whitespace and comments, takes approximately 170 lines of code.

Assignment Objectives

The purpose of this assignment is to demonstrate competence in the following skills.

- Program design
- Using command line parameters
- String manipulation
- File handling
- Tree manipulation

These tasks reflect all the subject objectives apart from objective 4.

As part of your subject workload assessment, it is estimated this assignment will take 22 hours to complete.

Queries

If you have a question, please contact the subject coordinator as soon as possible.

Gordon Lingard
glingard@it.uts.edu.au
9514-7935
Room 04.559, Building 10

However, for frequently asked questions a FAQ file will be put up. Please check this file before emailing the coordinator with a question.

`/pub/prprog/assign/assign3/faq.txt`

If serious problems are discovered the class will be informed and an update will be included in the following file

`/pub/prprog/assign/assign3/errata.txt`

Please keep a look out for this file.

PLEASE NOTE. If the answer to your questions can be found directly in any of the following

- subject outline
- assignment specification
- faq.txt
- errata.txt
- UTS Online discussion board
- DS & A web page

You will be directed to these locations rather than given a direct answer.

PLEASE NOTE. Please do not send email in HTML format or with attachments. They will not be read or opened. Only emails sent in plain text format will be read.

Assignment Submission

You will submit your program via the `submitass3` program. It is assumed that your C++ file is called `assign3.cpp` (although you may call it something else).

You must start in a directory that is within your home directory on `rerun` and then run

```
/home/glingard/submitass3 assign3.cpp
```

This assumes that `assign3.cpp` is in the current directory.

The `submitass3` program will then perform a number of tests. These include.

Compiling

Your program must compile without fatal errors or warnings – the `g++` compiler. Your program will be compiled with the following command and options

```
g++ -pedantic-errors -Wall -Werror
```

You can learn what these compiler options mean by running `man g++`.

PLEASE NOTE. Your program must compile on the student UNIX server. Programs written on Window's machines sometimes don't compile or run properly on the student server.

Style Feedback Program

Your program will be run through a style feedback program, which will check that your code meets a minimum style layout. If your program does not meet the standard a warning message will be printed. The style feedback program will display messages showing which lines of code do not conform to the standard and why they don't.

Code Feedback Program

Your program will be run through a code feedback program, which will check that your code meets minimum coding practices. If your program does not meet the standard a warning message will be printed. The code feedback program will display messages showing which lines of code are not acceptable and why they aren't.

Test Filter

Your program will be tested with 5 different tests. The results of your program will be compared to the results generated using the benchmark `spellcheck` program.

PLEASE NOTE. If you have to `exit` your program for any reason please `exit` with the value 0, or `return` from the `main` function with the value 0. Do not use any other value. This is necessary in order to make the test filter work correctly. Otherwise the test filter will say your program has failed the test.

PLEASE NOTE. Make sure the output from your program is **EXACTLY** the same as that from the benchmark executable. **ANY** deviation of the output from your program to that of the benchmark executable will cause the test to fail.

Plagiarism Agreement

You will be required to agree to a statement that the assignment is your own work and that you haven't given your code to anyone else.

If all goes well, `submitass3` will reply with a message saying you have successfully submitted the assignment. It will also place in your current directory a file called `receipt.txt`.

You may submit your assignment as many times as you like. The last assignment received will be the one marked.

PLEASE NOTE. Only assignments submitted via the `submitass3` program will be accepted for marking.

receipt.txt

receipt.txt is your proof that you have submitted your assignment. Once you have received it copy it to somewhere safe, such as your home directory. Additionally, copy your C++ file into the same location. Do not modify them in any way. If you wish to continue developing your program then do it on a duplicate file.

The receipt.txt file contains three pieces of information

1. A line saying you have submitted your file and when you did it.
2. A checksum of your C++ file
3. A checksum of your receipt

If you tamper with your C++ file or the receipt then the checksums will become invalid and therefore your receipt will become invalid. No actions will be taken if receipts have invalid checksums.

Acceptable Practice vs Academic Malpractice

- Students should be aware that there is no group work within this subject. All work must be individual. However, it is considered acceptable practice to adapt code examples found in the lecture notes, labs and the text book for the assignment. Code adapted from any other source, particularly the Internet and other student assignments, will be considered academic malpractice. The point of the assignment is to demonstrate your understanding of the subject material covered. It's not about being able to find solutions on the Internet.
- You should also note that assignment submissions will be checked using software that detects similarities between students programs.
- Do not let anyone submit their assignment from your account. The submitass3 program copies your assignment into a secure directory based upon your user login name. Anyone else using your account will have their assignment placed in your directory. Students who do this will find themselves reported to the Faculty for possible academic malpractice and misuse of Faculty resources.
- Participants are reminded of the principles laid down in the "Statement of Good Practice and Ethics in Informal Assessment" in the Faculty Handbook. Assignments in this subject should be your own original work. Any collaboration with another participant should be limited to those matters described in the "Acceptable Behaviour" section. Any infringement by a participant will be considered a breach of discipline and will be dealt with in accordance with the Rules and By-Laws the University. The Faculty penalty for serial misconduct of this nature is zero marks for the subject. For more information, see
- http://wiki.it.uts.edu.au/start/Academic_Integrity

Assignment Security

It is important to note that you have a responsibility to maintain the security of your assignment files. You can read more details about this in the plagiarism section of the DS & A web site - <http://learn.it.uts.edu.au/dsa/>

Assessment

Marks will be awarded out of 20 based upon the following scheme.

- Between 0 and 8 marks will be awarded by the computer based on the number of tests passed. The first 3 tests are worth 2 marks each while the final 2 are worth 1 mark each.

PLEASE NOTE. Some students have been known to write their code to artificially pass the tests rather than solve the assignment problem. In such cases a reduced mark will be given for the tests, including being given a 0.

The following marks will only be awarded if a score of 5 or more is awarded for the tests and both the code feedback and the style feedback programs are passed without generating warnings.

- Between 1 and 9 marks will be awarded on the quality of your code design and the algorithms used. This will also include looking at the proper use of `const`, etc.
- Between 1 and 3 marks will be awarded on the presentation style of the program. This involves meaningful variable names, intelligent use of comments and so forth.

PLEASE NOTE. It is a fundamental requirement of this assignment that you use the template binary tree to store words, as outlined in the requirement section. Students who use arrays to store the words will receive 0 for the assignment.

Preliminary Draft

The minimum requirements for the preliminary draft is as follows

- The program is getting a test score of 4+ upon submission. It is not required to pass the feedback programs.

PLEASE NOTE. Failure to have the preliminary draft in on time will mean that 4 marks will be deducted from your final mark. For example, if you leave doing the assignment till the final week, then the highest mark you can achieve is 16/20.

Late Assignments, Extensions and Special Consideration

Please read the subject outline regarding late assignments and extensions. If you did not get the outline a copy can be found at the DS & A web site

<http://learn.it.uts.edu.au/dsa/>

Assignments that are late by less than one week will incur a penalty of 1 mark for each day or part thereof late. Assignments more than a week late will not be accepted under any circumstance as the assignment solution will have been made available by then.

An extension of one week will only be granted if there is a fully documented reason which merits it. The documentation must be presented to the Subject Coordinator before the assignment due date. Extensions longer than a week will not be granted under any circumstance. If a one week extension is granted that means the assignment will be accepted up to a week after the due date without penalty. It will not be accepted later than that.

PLEASE NOTE. If something happens in the final week that prevents you submitting the assignment in on time and, you have not got the preliminary draft in by the draft due date, then no extension will be granted.

Students may apply for special consideration if they consider that illness or misadventure has adversely affected their performance in the assignment. For more information go to

http://www.sau.uts.edu.au/exams_ass/spec_cons.html

Return of Assessed Assignments

The code of your assignment will be printed out and marked. It will list codes for particular types and programming issues. These codes can be found in the DS & A web site in the

assessment->assignment->assessment menu section and will give a complete break down of your marks.

The marked assignments will be returned via the Student Center on level 2 of Building 10. The estimated return date is 17/6/11.

Getting Marks

You can check on your marks by running the following program.

```
/home/glingard/getmarks
```

Apart from your marks this program will give the following information.

- Given out – the date the assignment is given out.
- Due date – the date the assignment is due.
- Late date – assignments will be accepted up to one week after the due date but with a penalty of -1 mark per day or part thereof late.
- Marks released – The date the marks are released.
- Close date – The assignment is closed and the solution will be released. The close date will be one week after the assignments are marked and given back.

PLEASE NOTE. It is your responsibility to check that I have received your assignment and given you a mark. Even if you have a receipt you should check your mark and inform me if there is any problem before the close date. The `getmarks` program allows you to do this very easily. Unless you have a valid receipt or there are exceptional circumstances (e.g. serious medical conditions) no further correspondence regarding the assignment will be entered into after the close date.