Assignment 1: Textbuffer

jump to... assessment submission dryrun faq

Deadline	~~Friday 14 December, 11:59pm~~ Sunday 16 December, 11:59pm
Marks	100 marks (10 course marks)
Late Penalty	0.42 marks off the ceiling mark for each hour late. Submissions close **five days** after the deadline.
Submission	`2521 classrun give assign1`

A text buffer is an ordered collection of strings, where each string represents one line of text. Your task is to implement an abstract data type that meets the `Textbuffer` interface, and to implement tests for your implementation.

Get a copy of the provided files by running

```
 $ 2521 fetch assign1
'./textbuffer.h' -> '.../textbuffer.h'
./
Makefile
test_textbuffer.c
textbuffer.c
```

Do not modify `textbuffer.h`! If you do, your air-conditioning will mysteriously fail.

You need to implement the ADT defined in `textbuffer.h`; all your code should be in a file called `textbuffer.c`. You also need to construct a black-box test suite, in a file called `test_textbuffer.c`.

Assessment

The assigment is worth 100 marks, contributing 10% of your final course mark.

Implementation (automarked; 60 marks)
We will run your implementation against a number of tests to determine its completeness and correctness. These will be much more comprehensive than the tests we run during submission. You get marks for each test you pass.

Black-Box Testing (automarked + subjective; 10 marks)
You will create a suite of black-box tests in `test_textbuffer.c`, which will be run against some of our own correct and incorrect implemetations of textbuffers. Your tests should be able to pick up our errors, and pass our correct implementations.
Your tests will be automatically assessed on code coverage. Your tutor will also subjectively assess your tests.

White-Box Testing (automarked + subjective; 10 marks)
You will create a suite of white-box tests in `textbuffer.c`, in the function `white_box_tests` (with as many helper functions as you like).
Your tests will be automatically assessed on code coverage. Your tutor will also subjectively assess your tests.

Style (subjective; 20 marks)
Your tutor will subjectively assess your mechanical and semantic style, the appropriateness of your representation, and your algorithmic complexity.
Mechanical style deals with consistency of indentation, whitespace, layout. Semantic style deals with variable and function names, and the meaning conveyed in your implementation.

Style marking will also include assessing your choice of an appropriate representation for your ADT, and for the efficiency of the algorithms you choose. For example, you will lose marks if your implementation of a function has work complexity of $O(n^2)$, when there is a solution with $O(n)$ or $O(n \log n)$.

Bonuses: undo/redo, diff (automarked + subjective; 10 marks)
There are two optional challenges. Completion of these parts is optional.

Submission

To submit the assignment, execute the following command on a CSE machine:

```
$ 2521 classrun give assign1
```

You can check your submission by running

```
$ 2521 classrun check assign1
```

You will be able to collect your assignment once marked by running

```
$ 2521 classrun collect assign1
```

You may also submit, check, and collect from WebCMS 3.

You may submit multiple times; only the last submission will be marked. Submit partial solutions, and submit early!

Note that the submission system tends to be under high load close to assignment deadlines. We **will not** grant special considerations for lateness due to submission delays. You have been warned!

Unless otherwise stated, if you wish to submit an assignment late, you may do so, but a late penalty reducing the maximum available mark applies to every late assignment. The maximum available mark is reduced by 10% each day it is late and can be submitted a maximum of 5 days late.

Assignment extensions are only awarded for serious and unforeseeable events. Having the flu for a few days, deleting your assignment by mistake, going on holiday, work commitments, etc. do not qualify! You should aim to complete your assignments *well before* the due date, in case of last minute illness, and make regular backups of your work.

This is an individual assignment. You should develop your own solution without help from others. In particular, it is not permitted to exchange code or pseudocode. You are not allowed to use code developed by persons other than yourself. A reminder: plagiarism is an act of academic misconduct with severe penalties. See UNSW's resources on plagiarism and academic integrity.

If you have questions about the assignment, ask your tutor or a member of the course staff.