**COMP10002 Assignment 2 Feedback: abungla**

**Program Presentation**

Including: layout and style, readability, adherence to coding expectations, general care and appearance.
Some subset of the following lines will be retained by the marker.

- Up to Stage 1 initial allocation, +1.
- Up to Stage 2 initial allocation, +2.
- Up to Stage 3 initial allocation, +3.
- Up to Stage 4 initial allocation, +3.

*Deductions*

Some subset of the following lines will be retained by the marker. Marks in each section won't go below zero.

*Additions (included in total mark only if marks lost within this first section)*

- comment that says "algorithms are fun", +0.5;

*Other Comments from Marker*

**Structure and Approach**

Including: decomposition in to functions, development of data structures and declaration of them at the appropriate locations, choice of parameters to functions, use of appropriate `struct` and `typedef` declarations, choice of algorithms or methods.

Some subset of the following lines will be retained by the marker.

- Up to Stage 1 initial allocation, +1.
- Up to Stage 2 initial allocation, +2.
- Up to Stage 3 initial allocation, +3.
- Up to Stage 4 initial allocation, +4.

*Deductions*
Some subset of the following lines will be retained by the marker. Marks in each section won't go below zero.

- insufficient use of functions, -0.5;

*Other comments from marker*:

**Program Execution**

Including: compilation, execution on test data, output presentation and readability.
Programs that do not compile in the test environment will lose all marks in this section. Be sure to `verify` your submission and **check the output** before you say "finished" to yourself.

Some subset of the following lines will be retained by the marker.

- Up to Stage 1 initial allocation, +2.
- Up to Stage 2 initial allocation, +4.
- Up to Stage 3 initial allocation, +6.
- Up to Stage 4 initial allocation, +8.

*Deductions*

Some subset of the following lines will be retained by the marker. Marks in each section won't go below zero.

- deduct all execution marks if program generates no output at all; otherwise, deduct marks for the following items:

**Late Penalty**

If you make a late submission without having prior agreement from the lecturer, late submission penalties will apply. Overall marks won't go below zero.

**Total Mark: 14.5/15**

*Overall comments from marker*:

You've done well here. The program works as expected and passed our hidden test cases. The code structure, format and spacing was good. Your stage 3 function was a big long, I suggest creating another function that deals with making a new node and inserting it into the linked list. If I'm being honest, this assignment was hard to mark. Not because the code was hard to read, but there were a lot of instances that were very close to be able to penalise. I didn't end up choosing to penalise marks, but i'll list out the things I thought could've been done better. You want to create a separate struct to represent a node in a linked list, rather than keeping a next pointer in your transactions_t struct, and this is usually the preferred design choice. We want a node in any data structure to just contain another data object, and then pointers. Your algorithm in compare function seems a bit excessive, and you're copying a lot of data every time you compare. What if copying data was really expensive? The explanation for stage four complexity is good, but rather than using specific numbers to show a pattern to prove complexity, actually analysing the alogorithm would suffice. Talk more about if you're traversing the linked list of n nodes, with each node having to binary search up to m cards. Since binary search costs O(logn), searching m cards would cost O(logm)...

*Assignment Marker: Alec Yu*
*Subject Coordinator: Jianzhong Qi, jianzhong.qi@unimelb.edu.au*