**CS 260 Group Evaluation for Hash Table Lab**

|  |  |
| --- | --- |
| Your name:  (Person whose work is being evaluated) | Steven Wilson |
| Names of other group members participating in the evaluation: | Alexander Meyers |
| Date: | 2019/05/14 |

Instructions: You should have already completed the assignment and uploaded your solution files to Moodle. After you and another student (or students) have evaluated your work, you will submit this evaluation along with any revisions to your lab work to Moodle. You will be graded on your revised lab work and the quality of this evaluation, but this evaluation will not determine your grade.

|  |  |
| --- | --- |
| **Criteria** | **Evaluation** |
| ***String hash program*** |  |
| Does the program compile without errors or warnings and run without crashing? | Yes |
| Does the hash algorithm properly place items in the array? | Yes |
| Does the program properly delete items by marking them? | Yes |
| Does the program properly rehash items to the second table instead of copying them? | No |
| Does the program properly deal with deleted items when rehashing? | No |
| Does the program generate the proper output? | Yes |
| ***Word hash program with Separate Chaining*** |  |
| Does the program compile without errors or warnings and run without crashing? | Yes |
| Does the hash algorithm properly place items in separate slots? | Yes |
| Does the program properly delete items? | Not sure using std::list is okay |
| Did the program properly implement separate chaining? | Yes |
| Does the program generate the proper output? | Yes |

General Comments:

1. Your resize function needs to properly rehash the table, not just copy over elements. You do resize the array, but don’t set the new array size, so there is effectively no change.
2. I’m not sure the use of std::list from the standard library is okay here. The examples on Moodle show the implementation using nodes.

Other than those, the output looks pretty good, and it passes the tests as I see them.