**CS 3050 Homework # 2.** **Name : KEY**

**Submitted to Blackboard, due at 11:59pm on Feb. 15, 2018.**

1. Draw a picture of the sequence {S, T, R, A, I, G, H, T} stored as a doubly linked list using the multiple-array representation. Do the same for the single-array representation.

**Answer:**
*Using the format from Figure 10.5 from the book:*

| **Multiple-Array Representation** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Next | 1 | 2 | 3 | 4 | 5 | 6 | 7 | / |
| Key | S | T | R | A | I | G | H | T |
| Prev | / | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

| **Single-Array Representation** | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| K/N/P | S | 3 | / | T | 6 | 0 | R | 9 | 3 | A | 12 | 6 | I | 15 | 9 | G | 18 | 12 | H | 21 | 15 | T | / | 18 |

2. Give an efficient algorithm in pseudo code to reverse a singly linked list of n elements. The algorithm should use no more than constant storage beyond that needed for the list itself.

**Answer:** Many variations possible. Ensure students used a linked list and used proper syntax when performing operations with the linked list. The implementation should be able to perform the operation in-place without the need of any external data structures.

3.Given an Hash function $f(x) = x^2$ for 1000 integers in the range of –1000 to 1000, list 2 problems that could arises. Suggest a better Hash function to resolve these problems.

**Answer:** The first issue involves the negative numbers getting mapped to the same hash indices as the positive numbers. The second issue involves unused space, because the negative numbers are getting mapped into half the space, there is a lot of space not accessible.
A potential solution invovles any attempt to offset the output range by +1000, e.g.
$f(x) = x + 1000$

4. Demonstrate what happens when we insert the keys 3, 11, 80, 74, 92, 1024, 32, 59, 503, 293, 2010, 22, 104 into a hash table with collisions resolved by chaining. Let the table have 11 slots, and let the hash function be h(k) = k mod 11.

**Answer:**

| x | h(x) | Hashtable Contents |
|---|---|---|
| 3 | 3 | [0] : {22} => {11} |
| 11 | 0 | [1] :{1024} |
| 80 | 3 | [2] : |
| 74 | 8 | [3] : {80} => {3} |
| 92 | 4 | [4] : {59} => {92} |
| 1024 | 1 | [5] : {104} |
| 32 | 10 | [6] : |
| 59 | 4 | [7] : {293} |
| 503 | 8 | [8] : {2010} => {503} => {74} |
| 293 | 7 | [9] : |
| 2010 | 8 | [10] : {32} |
| 22 | 0 | |
| 104 | 5 | |

5. Consider a hash table of size m = 128 and a corresponding hash function h(k) = floor( m (kA mod 1) ) for A = (sqrt(5)-1)/2. Compute the locations to which the keys 1000, 1001, 1002, 1003, and 1004 are mapped.

**Answer:**

$h(1000) = floor\big(128(0.0339)\big) = 4$
$h(1001) = floor\big(128(0.6520)\big) = 83$
$h(1002) = floor\big(128(0.2700)\big) = 34$
$h(1003) = floor\big(128(0.8880)\big) = 113$
$h(1004) = floor\big(128(0.5061)\big) = 64$