**Take Home: Quiz 2 (15 pts) – Dynamic Singly Linked Lists**

Using Canvas, submit your quiz. You will submit your assignment in the ***lab*** Canvas space. Upload your solution to the appropriate “Quiz” space. You must upload your solution named <your last name>\_quiz2.pdf by the due date and time.

1. Write a function deleteAtPositionN() for a *singly-linked* list that has the following declaration and precondition:

int deleteAtPositionN (struct node \*\*pHead, int n,

int \*pData);

Precondition: n > 0.

The function should find the node at position *n*, and delete it, using function free(). The *data* should be returned indirectly through pData, then the node must be released back to the *heap*. The first node in the list starts at position 1. The function should return 1 if a node was deleted; 0 otherwise. Assume that struct node is defined as follows:

struct node

{

int data;

struct node \*pNext;

};

Solution:

int deleteAtPositionN(struct node\*\* pHead, int n, int\* pData)

{

int i = 1;

int success = 0;

if (\*pHead != NULL)

{

struct node\* pCur = \*pHead;

struct node\* pPrev = NULL;

while (pCur != NULL && i < n)

{

pPrev = pCur;

pCur = pCur->pNext;

i++;

}

if (pCur != NULL) //The position n exceeds the size of the list

{

success = 0;

}

else //The position n is pCur node

{

if (pCur != \*pHead)

{

pPrev->pNext = pCur->pNext;

}

else //pCur == \*pHead

{

\*pHead = pCur->pNext;

}

\*pData = pCur->data;

free(pCur); //Remove this node from the heap

success = 1;

}

}

return success;

}