**Short Answers - Using the Node class as presented in class (or the videos), answer the following questions.**

1. What is the one statement that can be used to insert a new node at the head of a linked list. Assume that the list's head\_pointer is called head\_ptr and the that the data for the new node is called entry.

Push();

1. Implement the following function as a new function for the linked list toolkit. (Use the usual node definition with member variables called data and link.)

int count\_42s(Node\* hPtr);

// Precondition: hPtr is the head pointer of a linked list.

// The list might be empty or it might be non-empty.

// Postcondition: The return value is the number of occurrences

// of 42 in the data field of a node on the linked list.

// The list itself is unchanged.

count =0; // initialize count to zero

for (list=hPtr; list!=NULL; list=list.getLink()) { //loop through each element of list

if (list.getData() == 42) { // if element is equal to 42

count++; // increment count

}

return count; // return count

}

1. Implement the following function as a new function for the linked list toolkit. (Use the usual Node definition with member variables called data and link.)

bool has\_42(Node\* hPtr);

// Precondition: hPtr is the head pointer of a linked list.

// The list might be empty or it might be non-empty.

// Postcondition: The return value is true if the list has at least

// one occurrence of the number 42 in the data part of a node.

for (list = hPtr; list !=NULL; list=list.getLink()) { //loop through elements in list

if (list.getData() == 42) { // if element matches 42

return true; // state as TRUE

} else { // or

return false; }// state as FALSE

}

1. Implement the following function as a new function for the linked list toolkit. (Use the usual Node definition with member variables called data and link. The data field is an int.)

int sum(Node\* hPtr);

// Precondition: hPtr is the head pointer of a linked list.

// The list might be empty or it might be non-empty.

// Postcondition: The return value is the sum of all the data components

// of all the nodes. NOTE: If the list is empty, the function returns 0.

sum = 0; // initialize sum to zero

for (list = hPtr; list != NULL; list=list.getLink()) { // loop through each element

if (list == NULL) { // if list has nothing

return 0; // end program successfully

} else { // or

sum = sum+list.getData(); // summation of data components

}

return sum; // return the sum and end function

}

**Multiple Choice**

1. Suppose cursor points to a node in a linked list (using the node definition with member functions called data and link). What statement changes cursor so that it points to the next node?
   * A. cursor++;
   * B. cursor = getLink( );
   * C. cursor += getLink ( );
   * D. cursor = cursor-> getLink ( );
2. Suppose cursor points to a node in a linked list (using the node definition with member functions called data and link). What Boolean expression will be true when cursor points to the tail node of the list?
   * A. (cursor == NULL)
   * B. (cursor-> getLink( ) == NULL)
   * C. (cursor-> getData( ) == NULL)
   * D. (cursor-> getData( ) == 0.0)
   * E. None of the above.
3. Suppose that f is a function with a prototype like this:

void f(\_\_\_\_\_\_\_\_ head\_ptr);

// Precondition: head\_ptr is a head pointer for a linked list.

// Postcondition: The function f has done some manipulation of

// the linked list, *and the list might now have a new head node.*

What is the best data type for head\_ptr in this function?

* + A. node
  + B. node&
  + C. node\*
  + D. node\*&

1. What is the expression for generating a pseudorandom number in the range 1...N?
   * A. rand() % N;
   * B. rand() / N;
   * C. rand() % (N + 1);
   * D. rand() / (N + 1);
   * E. (rand() % N) + 1;
2. Which expression computes a pseudorandom integer between -10 and 10 using rand() from <cstdlib>?
   * A. (rand( ) % 20) - 10
   * B. (rand( ) % 21) - 10
   * C. (rand( ) % 22) - 10
   * D. (rand( ) % 20) - 11
   * E. (rand( ) % 21) - 11