DEPT. OF ELECTRONICS AND
COMMUNICATIONS ENGINEERING
FACULTY OF ENGINEERING
CAIRO UNIVERSITY
GIZA, EGYPT

قسم الإلكترونيات و الاتصالات الكهربية
كلية الهندسة
جامعة القاهرة
الجيزة – جمهورية مصر العربية

Third Year --- Computers: Data Structures
March 2011

Sheet 2: Lists ADT

1) Write a program that uses the linear list class to store 100 integer numbers. Add two functions to the class:

i- int List.search(elementtype & elem) to search for certain element in the list.
ii- void List.sort( ) to sort the items of the list.

Write a driver program that reads the numbers and adds them to the list, then uses the sort function to sort the numbers and prints the list after sorting.

2) Write a program that creates an inorder linked list of strings, reads in each word from an input file, puts them into the inorder list, and then prints a table of word frequencies, listing each word that occurs more than once.

3) To the standard linked list implementation, add the following functions:

i- void L.append(List l) glues list l to the end of this list(L)
ii- List* L.copy() makes a copy of this list and returns a pointer to it.

4) Add a reverse operation to the standard linked list implementation. Implement the reverse operation recursively, using the following observation: If the list is empty, do nothing. To reverse a non-empty list, each node should point to the node that was previously its predecessor, the header should point to the last node, and the node that was formerly first should have a null link.

5) Using the inorder linked list implementation, add a merge function:

void L.merge(List & mergeList);

Precondition: List and mergeList are lists in sorted order.
Postcondition: After merge, List contains a merge of the items in List and mergeList.(mergeList remains unchanged).

6) Modify problem (5) by adding a sort function to the standard class linked list to sort the two lists before merging them.

7) To the standard linked list, add a function *remove* to delete any item from the linked list and returns the deleted node to the memory. Then use the class linked list to delete all odd elements (first, third, fifth, ...).

8) Write a function that counts the number of the nodes in a linked list.

9) Write a function "insertBefore" that will insert an element before the current node of a linked list, make the necessary correction to the list.

10) Modify problem (9) using the doubly linked list class.

11) **Programming Assignment**: Add the following function to the linked list class:
void L.Split(List & L1,List & L2); which splits this list (L) into two lists(L1 and L2)
Preconditions: Lists L1 and L2 are empty lists.
Postcondtions: L1 contains the first, third, fifth...etc nodes and L2 contains the second, fourth, sixth...etc nodes. Then write a program that adds 20 elements to the list L and then calls the function split. Test the output of the split function.