SET 3

Theme: Linked List. Representation
Operations. Design subalg. based on representation.

1. Dynamic Singly Linked List. Representation
(SLNode, Position, SLList).

2. Dynamic Singly linked list. Representation.
(DLNode, Position, DLList).

3. a) create an empty SLList
b) Insert an element in the head of a SLList.
c) InsertAfter in a SLList.
Specif & pseudocode
d) insert in a SLList – to combine insertHead with insertAfter
Specif & pseudocode
e) Given a sorted SLList (according to operator<), insert a new element and keep it sorted. Use a subalg. insertAfter

4. RemoveAll elements from a SLList. (clear it!)

5. Reverse the order of elements in a SLList.
Elements keep their positions. (Re-set only links between nodes.)

```
++STL
void reverse();
Reverses the order of elements in the list.
All iterators remain valid and continue to point to the same
elements.
This function is linear time.
```

6. Merge 2 sorted SLList of TCE (according to operator<). Elements keep their positions. Re-set only links between nodes.(pseudocode)
a) The original lists will not exists after merging.
b)similar with:

```
C++STL list
void merge (list& x);
Merges x into the list by transferring all of its elements at
their respective ordered positions into the container (both
containers shall already be ordered).
```

7. DLList
a) write a subalg. that interchanges two nodes in a list, by re-setting the links between nodes
b) Sort a DLList

- according to operator<
- pseudocode; use a

```
C++STL list
void sort()
Sorts *this according to operator<.
All iterators remain valid and continue to point to the same
elements.
```