

CPSC 131 Homework 5

Deadline: Monday, October 22 (Mon, Wed sections)
Tuesday, October 23 (Tue, Thu sections)

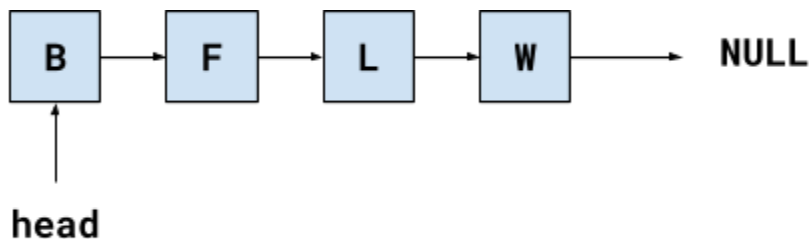
Turn in your submission as hard copy in class. Complete all homework problems but only one question will be graded.

Refer to your instructor's syllabus addendum to see their policy on group work. Some instructors allow homework to be completed in groups. Include all group member names on the homework. Only one homework should be turned in per group.

#1

Assume we have a single-linked char list that is terminated with a null address. Write a recursive function to get the last char value in the list. The drawing and given code should help.

Implement `char SinglyLinkedList::getTail(Node* ptr);`



```
struct Node {
    char data;
    Node* next;
};

char SinglyLinkedList::getTail() {
    return getTail(head);
}

char SinglyLinkedList::getTail(Node* ptr) {
    // YOUR CODE GOES HERE
}
```

#2

Assume we have a `DynamicVector` class that when it fills to capacity and needs more space we call a function `reserve(int)` which will do the following:

- 1) Reserve a new larger capacity (if it's actually larger)
- 2) Copy the previous values over to the new array
- 3) Delete the old array
- 4) Make sure our `DynamicVector` uses the new array

The following forward declaration with all of the `DynamicVector`'s member variables should help.

Implement `void DynamicVector::reserve(int newCapacity)`

```
class DynamicVector {
private:
    int capacity;           // current max capacity
    int size;               // number of elements in array
    char* array;            // array storing elements
    void reserve(int);      // new function to write
};

void DynamicVector::reserve(int newCapacity) {
    // YOUR CODE GOES HERE
}
```

#3

Sketch the contents of a stack built using a dynamic vector (as in Problem #2) after each of the statements below. Assume that the initial size of the array is 1. Show the contents of the array pointer and the "top" index (int t). Be sure to sketch out any empty capacity as well.

```
DynamicStack ds;
ds.push('A');
ds.push('B');
ds.push('C');
ds.push('D');
ds.pop();
ds.push('E');
ds.push('F');
```