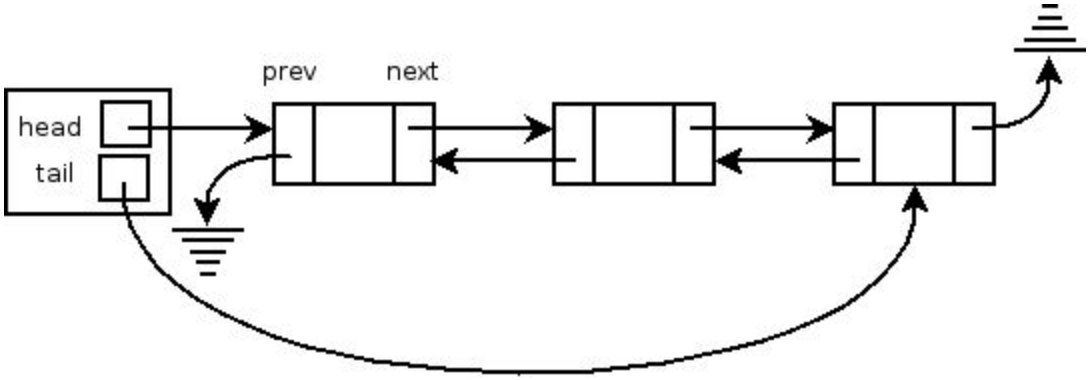
Homework 08 - Doubly Linked List
Due Sunday 11/18/2018 11:59pm

What to get before you start
Download hw8handout.tar file from Canvas. It will include the following files.

```
hw8handout/doublylinkedlist.c
hw8handout/doublylinkedlist.h
hw8handout/main.c
hw8handout/Makefile
hw8handout/mymalloc.c
```

Doubly linked list
Doubly linked list is a list where each node in the list has two pointers, **next** pointing to the next node and **prev** pointing to the previous node. A doubly linked list looks like this in the memory. Notice that the prev pointer of the head node and next pointer of the tail node points to NULL.



What to do

1. See the definition of data structures already provided in doublylinkedlist.h file. The main driver is also provided in main.c. You only need to implement the following two functions in doublylinkedlist.c file. Make sure you handle the edge cases properly.

    a. `void insertHead(doublyLinkedList *listPtr, int value);`
    Insert a new node that has value given to the head of the list. Make sure you update the head pointer contained in the doublyLinkedList structure. If this node is inserted into an empty list, you should also update the tail pointer in the doublyLinkedList structure.
    b. `int removeTail(doublyLinkedList *listPtr);`
    Remove the tail node from the doubly linked list and return the value it contains from the function. Remember to update the tial pointer contained in the doublyLinkedList structure. Make sure you use free to release the memory holding this node.

You will only need to make changes to **doublylinkedlist.c** file. DO NOT Modify any other files in the handout folder or add or remove any files or change any files' names in the folder. When we test your code, we will only copy over your doublylinkedlist.c file to see if it works with the rest of the files provided to you.

When you test the functions of your code, after each insert/remove, use both showhead and showtail command to print the list in both directions to make sure that the whole list is connected properly in both directions. Also test your code inserting into empty list, or removing an element from a list with just one node among other things. Always print the list from head and tail to see if all the links are in place.

What to submit
Submit just **doublylinkedlist.c** file.