7SENG010W Data Structures and Algorithms
Week 3 Tutorial Exercises: Linked Lists

Exercise 1 (b) Solutions

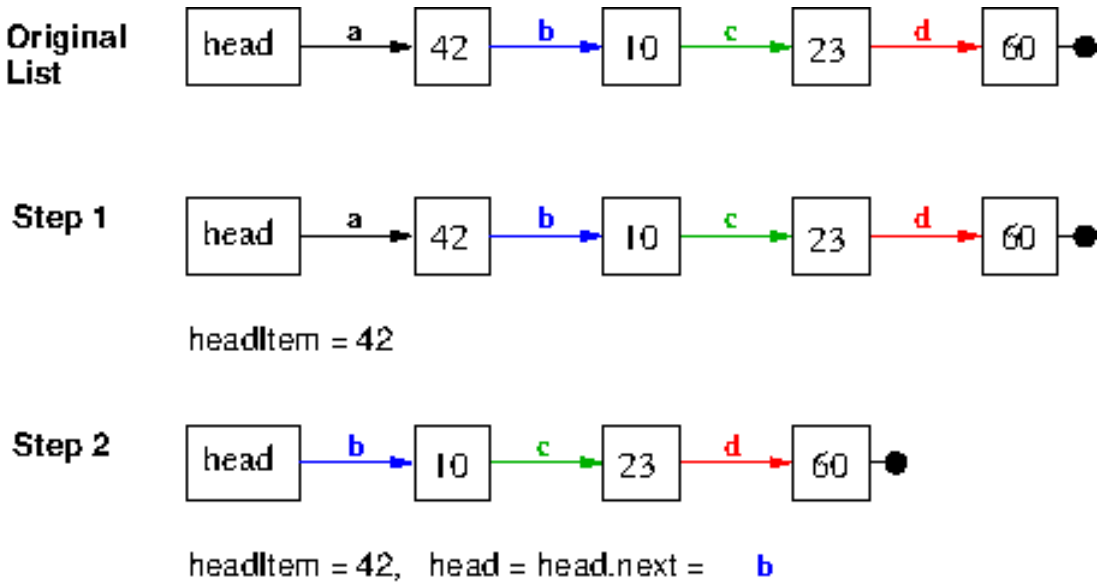
Using the list from Lecture 3 - < 42, 10, 23, 60 >, draw "List" diagrams in the style of Week 03 Lecture for the following operations:

deleteHead( ) - delete the head node if it exists & return head item, or null if empty
e.g. result is < 10, 23, 60 >

There are 2 steps for doing this operation:

Step 1: store the current value of the head node, i.e. 42, so that it can be returned.

Step 2: Set head = head.next, i.e. replace head's link **a** with 42's next link "**b**".



When the list is not empty (head != null) then return the vallue of headItem = 42,
if the list is empty (head == null) then there is nothing to do apart from returning headItem = null.

insertAtTail(99) - add an item to the tail of the list, e.g. result is < 10, 23, 60, 99 >

There are 4 steps for doing this operation:

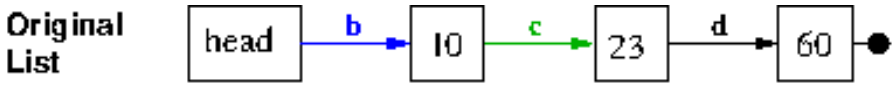
Step 1: find the current tail node, i.e. link **d** with item 60.

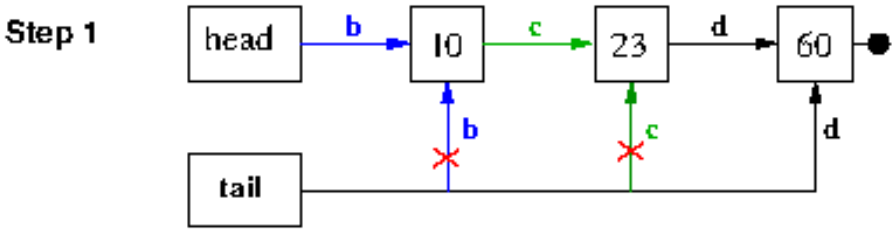
Step 2: create a new node to store the new tail value 99, i.e. newTail with link **e** & value 99.

Step 3: add the new tail (99) to the end of the current list by setting tail.next = newTail, i.e. replace 60's next **null** link with newTail's link **e**.

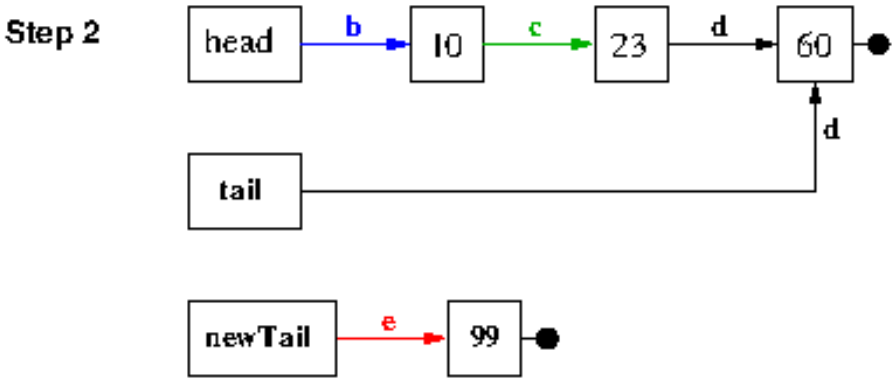
Step 4: completed the insertion of the new tail 99.

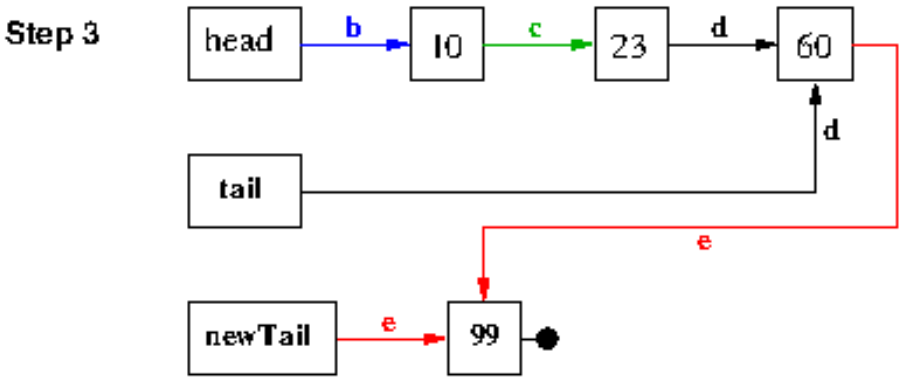
insertAtTail(99) List diagrams:





tail = d (60)





tail.next = newTail = e