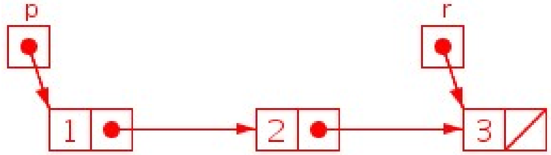
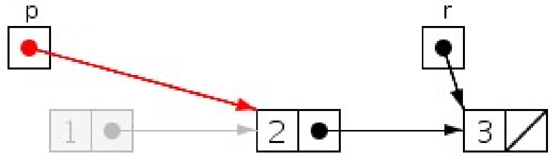
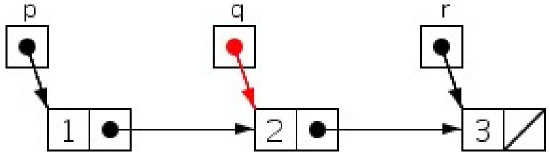
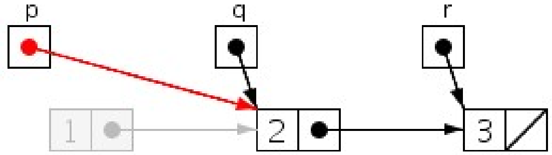
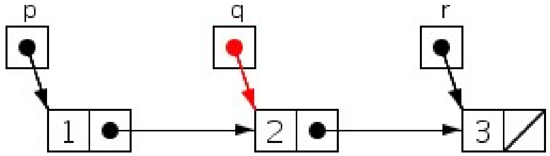
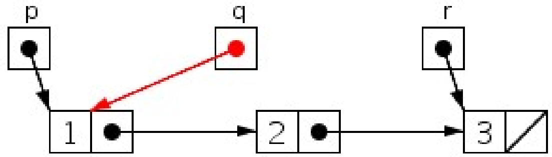
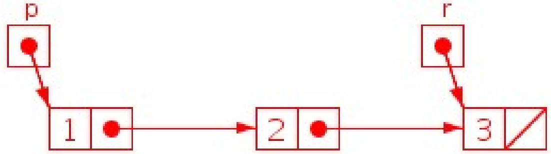
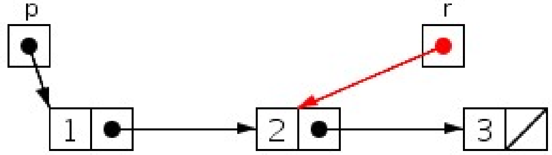
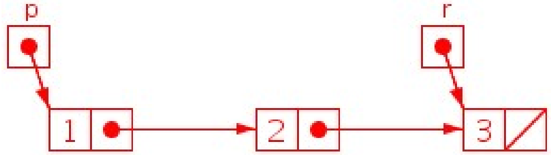
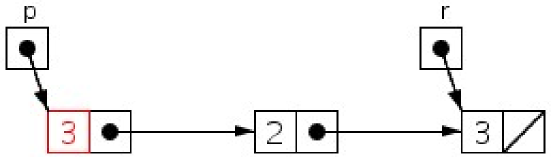
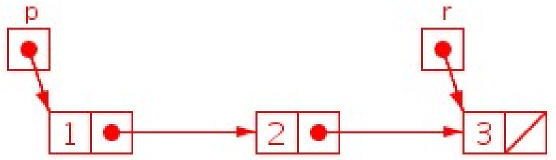
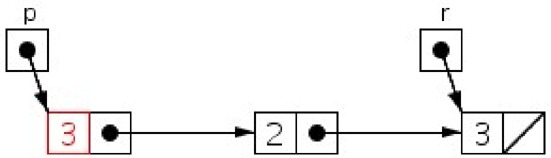
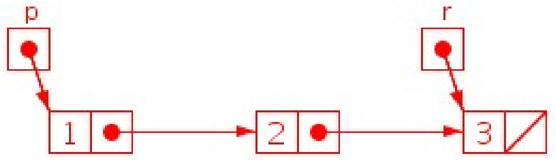
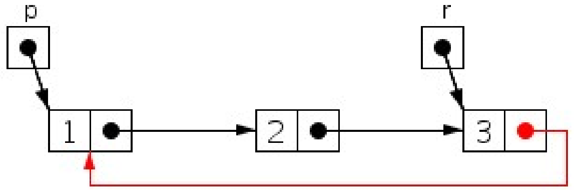
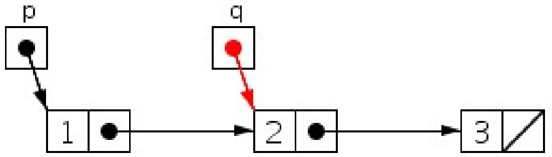
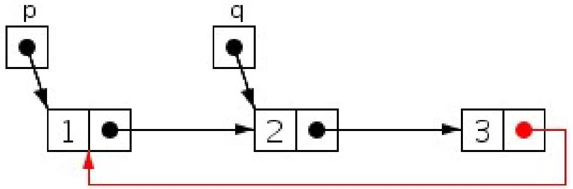
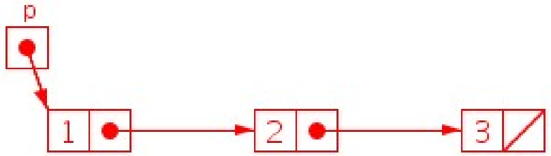
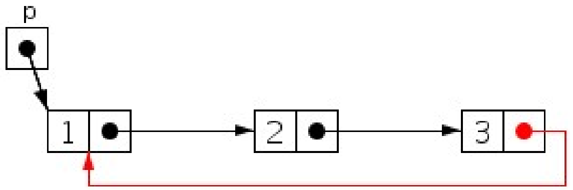
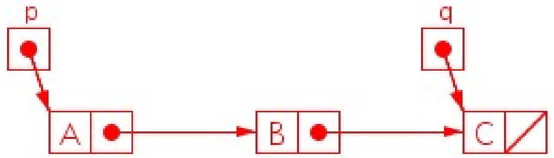
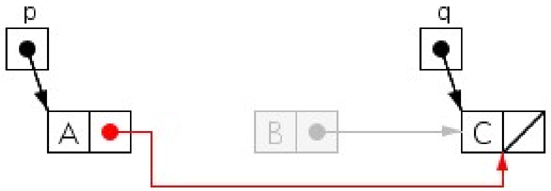
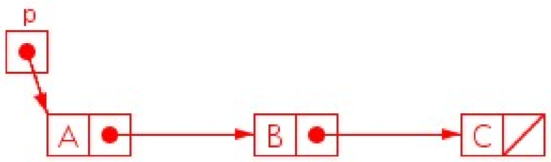
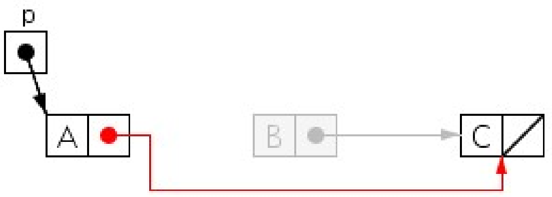
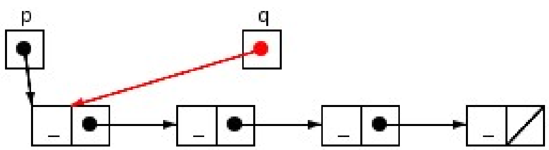
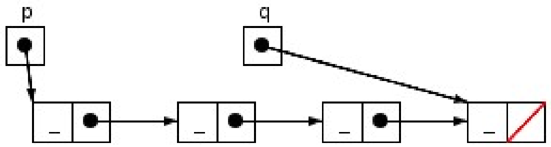
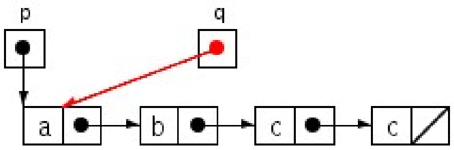
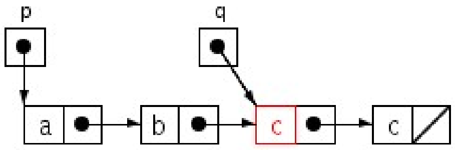
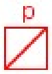
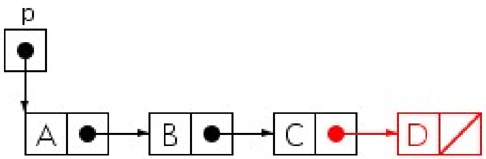
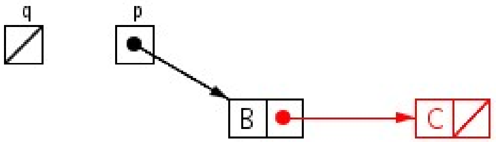
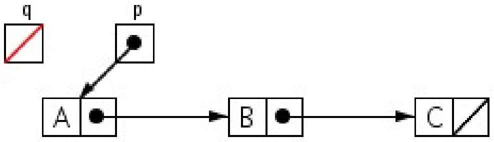
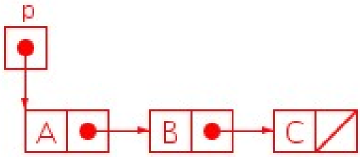
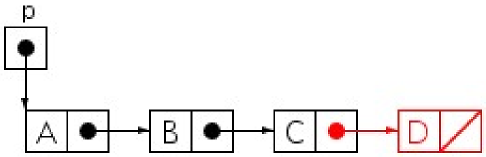
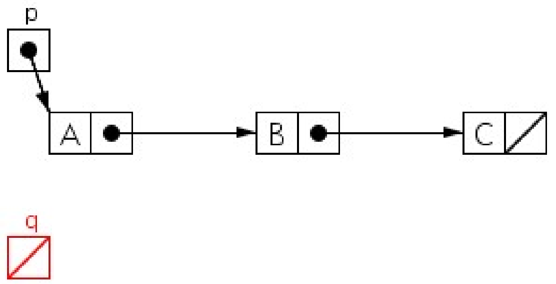
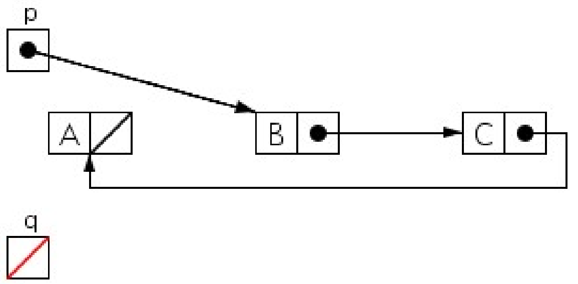
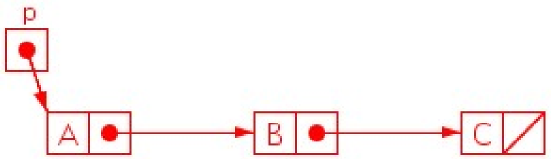
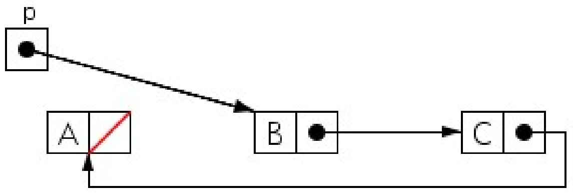
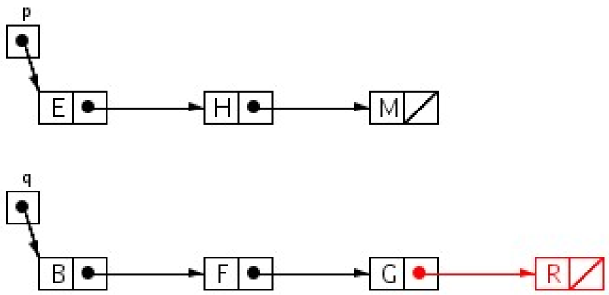
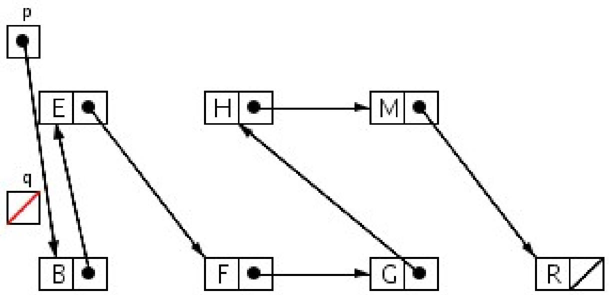
Linked list manipulations

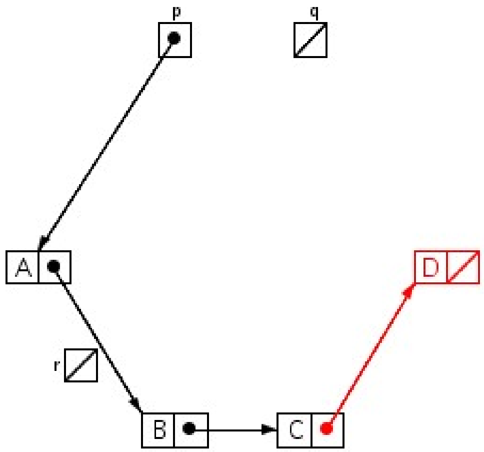
The first 20 exercises on this page are quite short. They will help you master basic linked list operations. The last 2 programming assignments are more challenging. They will help you become comfortable designing and implementing robust algorithms to manipulate linked lists.

	Initial Setup	Exercise	Final Configuration
1		Use a single assignment statement to make the variable p refer to the Node with info '2'	
2		Redo exercise 1 but, this time, your assignment statement *must* refer to both variables p and q.	
3		Use a single assignment statement to make the variable q refer to the Node with info '1'.	
4		Use a single assignment statement to make the variable r refer to the Node with info '2'.	
5		Use a single assignment statement to set the info of the Node referred to by p equal to the info of the Node referred to by r (you must access this info through r; do not refer to the character '3' directly).	

6		Redo exercise 5 by referring only to variable p (not to variable r). Again, you may *not* refer to the character '3' directly .	
7		Write a single assignment statement to transform the linked list headed by p into a *circular* linked list. Your assignment statement *must* refer to both variables p and r.	
8		Redo exercise 7 but, this time, your assignment statement *must* refer to both variables p and q.	
9		Redo exercise 7 but, this time, your assignment statement must refer *only* to variable p.	
10		Write a single assignment statement to remove the Node with info 'B' from the linked list headed by p. Your assignment statement *must* refer to both variables p and q.	

11		Write a single assignment statement to remove the Node with info 'B' from the linked list headed by p.	
12		Write a while loop to make q refer successively to each Node in the linked list headed by p. q must end up referring to the last Node in the list.	
13		Write a while loop to make q refer successively to each Node in the linked list headed by p until q refers to the first Node with info (lowercase) 'c'.	
14		Use four assignment statements, each referring to variable p, to create a linked list headed by p and containing 4 Nodes with info 'A', 'B', 'C', and 'D', in this order.	
15		Create a new Node with info 'A' and insert it at the beginning of the list headed by p.	
16		Create a new Node with info 'D' and insert it at the end of the list headed by p.	

17		Remove the Node at the beginning of the list headed by p and insert it at the end of the same list. Your program *must* refer to both variables p and q.	
18		Redo exercise 17 but, this time, your program must *only* refer to variable p.	
19		Merge the two lists headed by p and q into a single list headed by p in which the Nodes are sorted in alphabetical order.	

20		Using only the three existing variables p, q, and r, reverse the order of the Nodes in the list headed by p.