Tutorial 1

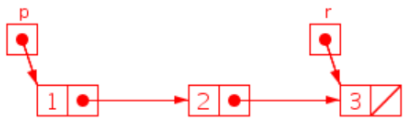
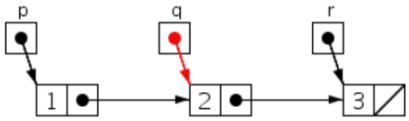
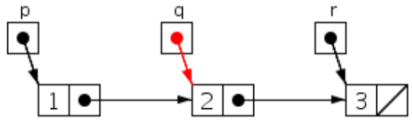
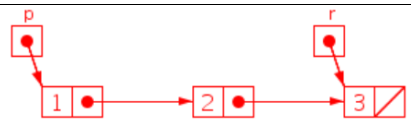
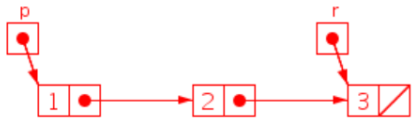
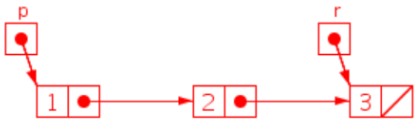
Topic Linked List

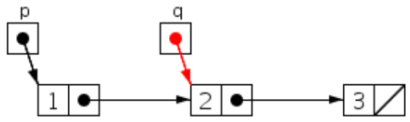
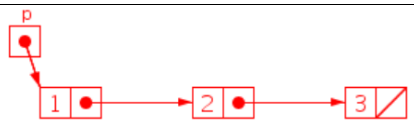
Use this declaration of the Node class:

```
struct Node {
    int info;
    Node * next;
};

Node *p, *r, *q;
```

Question 1

Initial Setup	*Exercise*	*Final Configuration*
	*Use a single assignment statement to make the variable p refer to the Node with info '2'*	*Example* ***p=p->next->data;***
	*Use a single assignment statement to assignment statement must refer to both variables p and q.*	*Code:* *p=p->data* *q=q->data*
	*Use a single assignment statement to make the variable q refer to the Node with info '1'.*	*Code:* *q=p->data*
	*Use a single assignment statement to make the variable r refer to the Node with info '2'.*	*Code:* *Code; r=p->next->data*
	*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)*	*Code:* *P-> data=r->data*
	*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.*	*Code:* *R-> next=p*

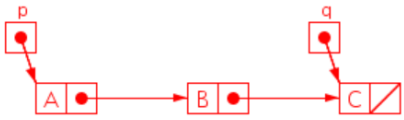
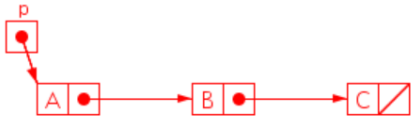
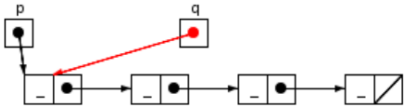
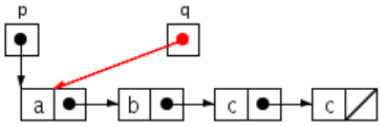
	*Write a single assignment statement to transform the linked list headed by p into a circular linked list. Your assignment statement must refer to both variables p and q.*	*Code:* *q->next->next=p*
	*Write a single assignment statement to transform the linked list headed by p into a circular linked list. Your assignment statement must refer only to variable p.*	*Code:* *P->next->next->next=p*

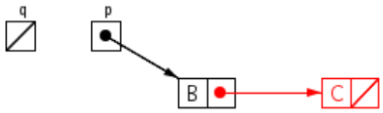
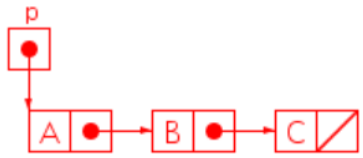
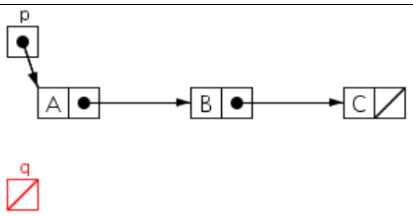
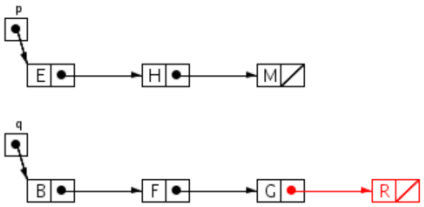
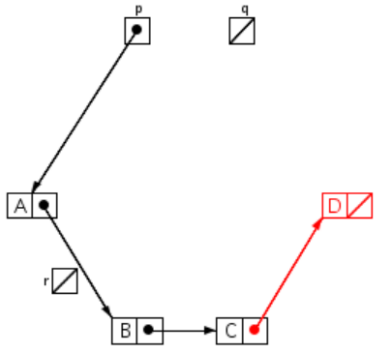
Question 2

Use this declaration of the Node class:

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


Node *p,*q;
```

Initial Setup	*Exercise*	*Final Configuration*
	*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.*	*Code:* *P->next=q*
	*Write a single assignment statement to remove the Node with info 'B' from the linked list headed by p*	*Code:* *p->next=p->next->next*
	*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.*	*Code:* *While (q==NULL)* *{* *q=p->next->next->next* *}*
	*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'.*	*Code:* *While (q==NULL)* *{* *q=p->next->next->info* *}*

	*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.*	*Code:* *p->info=A* *p->next->info=B* *P-> next->next->info=C* *p->next->next->next->info=D*
	*Create a new Node with info 'A' and insert it at the beginning of the list headed by p.*	*Code:* *Node*temp=newnode();* *Temp->info='A'* *Temp->next=head* *Head=temp*
	*Create a new Node with info 'D' and insert it at the end of the list headed by p.*	*Code:* *Node*temp=newnode();* *Temp->info=D* *p->next->next->next->temp* *temp->next=NULL*
	*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.*	*Code:* *P=temp* *P=p->next* *P->next->next->next->=temp* *Temp->next=NULL*
	*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.*	*Code:* *If (h1->data <h2->data)* *{* *h 1 ->next=merge (h1->next,h2);* *return h1;* *}* *else {* *h2->next=merge(h1,h2->next);* *return h2;*
	*Using only the three existing variables p, q, and r, reverse the order of the Nodes in the list headed by p.*	*Code:* *Node*temp=head;* *Node*prev=NULL, node*next=NULL;* *While (temp!=NULL)* *{* *Next=temp->next* *Temp->next=prev* *Prev=temp* *Temp=next* *}*