1)Insertion at the beginning of the linked list

Step 1:Check if the node is empty .If it’s empty terminate the process  
Step2:Create a new node  
Step 3:Update the pointer of new node to point to the current head  
Step 4:Update the head pointer to point at new node.  
Step 5: Stop

2)Insertion at the specified node of the linked list.

Step1:Check if the specified node exists.  
Step2:Create a new node  
Step3:Update the pointer of new node to point to the node that comes after specified node.  
Step4:Update the pointer of the specified node to point at the new node.  
Step5:Stop.

3)Insertion at the end of the linked list.

Step1:Create a new node.  
Step2:Update the pointer of last node to point to the new node.  
Step3.Update the pointer of new node to point to null.  
Step4:Stop.

4)Deletion at the beginning of the linked list.

Step1:Update the head pointer to point to the node after first node.  
Step2:Delete the old node to free up space.  
Step3:Stop.

5)Deletion at the specified node of the linked list

Step1:Find the node that you have to delete.  
Step2:Update the pointer of previous node to point to the next node of specified node.  
Step3:Delete the old node to free up space.  
Step3:Stop.

6)Deletion at the end of the linked list.

Step1:Find the the second last node.  
Step2:Update the pointer of the second last node to point at null.  
Step3:Delete the old node to free up space.  
Step4:Stop.

7) Insertion in stacks

Step 1: Check for stack overflow (Give error message if stack is full)  
Step2: Increment the top pointer (or index) of the stack.  
Step3: Place the 'element' at the position indicated by the updated top pointer (or index).  
Step4:Stop.

8)Deletion in stacks

Step 1: Check for stack underflow.(If the stack is empty give an error)  
Step 2: Remove the top element from the stack.  
Step 3: Decrement the top pointer  
Step 4:Stop.

9)