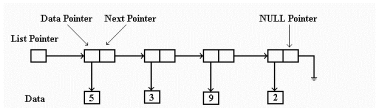
Linked Lists

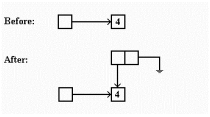
- A linked list is an ordered data structure (not necessarily sorted) that consists of a series of nodes.
- Each node contains:
  (1) A pointer to the data it represents.
  (2) A pointer to the next node in the list.
- Each list has a list pointer that points to the first node in the list. (The next element of the last node is usually set to NULL (0) to mark it as special.)



The fundamental operations on a linked list are:

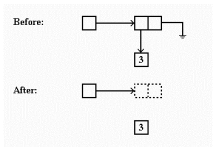
Allocate Node

Given a pointer to a block of data, Allocate Node creates a node for the linked list.



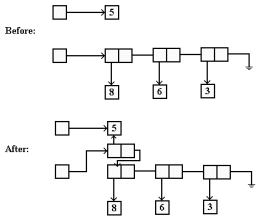
Free Node

Given a pointer to a node, Free Node destroys the node and returns its memory back to the free memory heap.



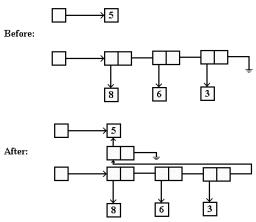
Insert Item

Given a pointer to a block of data, Insert Item creates a node and inserts into the front of the linked list.



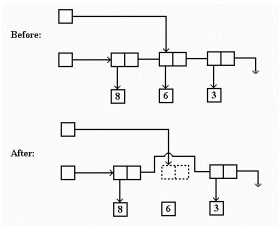
Append Item

Given a pointer to a block of data, Append Item creates a node and appends it to the end of the linked list.



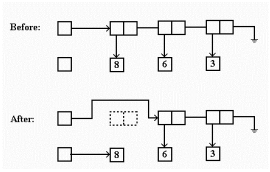
Delete Item

Given a pointer to an item, Delete Item deletes the node from the linked list and frees the deleted node.



Remove Item

Remove Item removes the first node from the linked list, and returns a pointer to its block of data.




```
class Node{
public:
        Point *pointData;
        Node *nextNode;
};

class List{
        Node* start_of_list;
        Node* end_of_list;
public:
        List(){start_of_list = NULL;
                end_of_list = NULL;}
        Node *AllocateNode(Point* pData);
        void FreeNode(Node* pNode);
        void InsertItem(Point* pData);
        void AppendItem(Point* pData);
        void DeleteItem(Node *pNode);
        Node* RemoveItem();
};
```