```
#include <iostream>
#include <cstdlib>
using namespace std;
 class Node // class is made with name node
{
public:
    int data;
    Node*next;


    Node(int x, Node* addr) // creation of a new node
{
        data = x;
        next = addr;
    }
};

class LinkedList // linked list class is created
{
    private:
    Node*       head;

    public:

    LinkedList() // the head of the linked list is initilized to null
 {
        head = NULL;
    }

    bool is_empty() // check if the list is empty
{
        if(head == NULL) return 1;
        else return 0;
    }

    // add an item to a linked list
    void node_add(int val) // element is added to the linked list
 {
        if(head == NULL) head = new Node(val, NULL);
        else {
            Node* n = head;
            while(n->next != NULL) n = n->next;
            n->next = new Node(val, NULL);
        }
    }



    void print() // print() function prints the linked list onto the
output.
{
        cout<<"[";
        if(head != NULL) {
            if(head->next == NULL) cout<<head->data;
            else {
                Node* n = head;
                while(n->next != NULL) // checking till the end of the
list
{
                    cout<<n->data<<",";
```

```
                    n = n->next;
                }
                cout<<n->data;
            }
        }
        cout<<"]\n";
    }
    void reverse() // reverse() function to reverse the linked list
{
        if((head != NULL)*(head->next != NULL)) {
            Node* m = head;
            Node* n = head->next;
            Node* t = NULL;
            if(n->next != NULL) t = n->next;
            head->next = NULL;
            while(t->next != NULL) {
                n->next = m;
                m = n;
                n = t;
                t = t->next;
            }
            n->next = m;
            head = t;
            t->next = n;
        }
    }
};

int main(void) // main () function
{

    LinkedList      L;
    unsigned int    i;

    srand(time(NULL));
    for(i=0;i<8;i++) L.node_add(rand() % 100); // linked list taking 8
random values using randon no. generator function

    L.print(); // calling the print () function

    L.reverse(); // calling the reverse() function to reverse the linked
list

    cout << " The reversed link list is :\n\n";

    L.print(); //again calling the print()  function to output the linked
list after reversal


    return(0);
}
```

```
/*Reverse linked list using recursion*/


 #include<iostream>
```

```
using namespace std;
struct node
{
    int data;
    struct node *next;
};
void insert(struct node **head,int x)
{
    struct node *newnode= new node[sizeof(node)];
    newnode->data = x;
    newnode->next = *head;
  *head = newnode;
}
void display(struct node *head)
{
    while(head!=0)
    {
        cout<<head->data<<"->";
        head=head->next;
    }
    cout<<endl;
}
struct node * reverse(struct node* newnode,struct node **head)
{
    if(newnode->next==0)
    {
       *head = newnode;
        return newnode;
    }
    else
    {
        struct node *prev = reverse(newnode->next,head);
       prev -> next = newnode;
       newnode -> next = 0;
        return newnode;
    }
}
int main()
{
  struct node *head = 0;
  int n=10;
  while(n!=0)
      insert(&head,n--);


    display(head);

    reverse(head,&head);
  display(head);
    return 0;
}
```