```c
//Singly linked list with header node
typedef struct node *Nodeptr;
struct node{
    int data;
    Nodeptr next;
};

Nodeptr getnode(){
    Nodeptr temp;
    temp = (Nodeptr) malloc(sizeof(struct node));
    return temp;
}

int IsEmpty(Nodeptr head){
    return (head->next==NULL ? 1:0);
}
void InsertFront(Nodeptr head, int x){
    Nodeptr temp;

    temp= getnode();
    temp->data = x;

    temp->next = head->next;
    head->next = temp;

}
```

```c
void Display(Nodeptr head){
    Nodeptr temp;

    if (IsEmpty(head)){
        printf("Empty List");
        return;
    }
    temp=head->next;
    printf("Contents of List : \n");
    while(temp){
        printf("%d\n",temp->data);
        temp = temp->next;
    }
}
```

```
void InsertLast(Nodeptr head, int x){
    Nodeptr temp, rear;

    temp= getnode();
    temp->data = x;
    temp->next = NULL;


    rear = head;
    while(rear->next)
        rear = rear->next;
    rear->next = temp;

}
```

```c
int DeleteFront(Nodeptr head){
    Nodeptr first;
    int x;

    if (IsEmpty(head)){
        printf("Empty List\n");
        return -999;
    }

    first = head->next;
    x = first->data;
    head->next = first->next;
    free(first);
    return x;
}
```

```
void Reverse(Nodeptr head){
    Nodeptr p,q,r;

    p= head->next;
    q=NULL;

    while(p){
        r=q;
        q=p;
        p=p->next;
        q->next = r;
    }

    head->next = q;
}
```

```c
int main(){
    Nodeptr head;
    //create a header node
    head = getnode();
    head->next = NULL;

    InsertFront(head,20);
    InsertFront(head,10);
    Display(head);
    InsertLast(head, 40);
    Display(head);
    DeleteFront(head);
    Display(head);
    Reverse(head);
    Display(head);
    return 0;
}
```