```
//Created By Ritwik Chandra Pandey on 03/03/21
//183215
//Reverse LL

#include <stdio.h>
#include <stdlib.h>

/* Link list node */
struct Node {
    int data;
    struct Node* next;
};

/* Function to reverse the linked list */
static void reverse(struct Node** head_ref)
{
    struct Node* prev = NULL;
    struct Node* current = *head_ref;
    struct Node* next = NULL;
    while (current != NULL) {
        // Store next
        next = current->next;

        // Reverse current node's pointer
        current->next = prev;

        // Move pointers one position ahead.
        prev = current;
        current = next;
    }
    *head_ref = prev;


}

void addNodes(struct Node **first, int x) {
    struct Node* temp= (struct Node*)malloc(sizeof(struct Node));
    struct Node* lastNode = *first;
```

```
    temp -> data = x;
    if (*first == NULL) {
        *first = temp;
    } else {
        while (lastNode -> next != NULL) {
            lastNode = lastNode -> next;
        }
        lastNode -> next = temp;
    }

}


/* Function to push a node */
void push(struct Node** head_ref, int new_data)
{
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    new_node->data = new_data;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

/* Function to print linked list */
void printList(struct Node* head)
{
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
}

/* Driver code*/
int main()
{
    /* Start with the empty list */
    struct Node* head = NULL;
    int val;
    do{
```

```
        printf("Enter an element to add elements : -1 to stop  ");
        scanf("%d",&val);
        if(val!=-1)
        addNodes(&head,val);
    }while(val!=-1);


    printf("Given linked list\n");
    printList(head);
    reverse(&head);
    printf("\nReversed Linked list \n");
    printList(head);

}
```