```c
// C program to reverse a linked list in groups of given siz
#include<stdio.h>
#include<stdlib.h>

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

/* Reverses the linked list in groups of size k and returns
pointer to the new head node. */
struct Node *reverse (struct Node *head, int k)
{
    if (!head)
        return NULL;

    struct Node* current = head;
    struct Node* next = NULL;
    struct Node* prev = NULL;
    int count = 0;



    /*reverse first k nodes of the linked list */
    while (current != NULL && count < k)
    {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
        count++;
    }

    /* next is now a pointer to (k+1)th node
    Recursively call for the list starting from current.
    And make rest of the list as next of first node */
    if (next != NULL)
    head->next = reverse(next, k);

    /* prev is new head of the input list */
    return prev;
}

/* UTILITY FUNCTIONS */
/* Function to push a node */
void push(struct Node** head_ref, int new_data)
{
    /* allocate node */
    struct Node* new_node =
            (struct Node*) malloc(sizeof(struct Node));

    /* put in the data */
    new_node->data = new_data;

    /* link the old list off the new node */
    new_node->next = (*head_ref);

    /* move the head to point to the new node */
    (*head_ref) = new_node;
}

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

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

    /* Created Linked list is 1->2->3->4->5->6->7->8->9 */
    push(&head, 9);
    push(&head, 8);
    push(&head, 7);
    push(&head, 6);
    push(&head, 5);
    push(&head, 4);
    push(&head, 3);
    push(&head, 2);
    push(&head, 1);

    printf("\nGiven linked list \n");
    printList(head);
    head = reverse(head, 4);

    printf("\nReversed Linked list \n");
    printList(head);

    return(0);
}

```