// C program to reverse a linked list in groups of given size

#include<stdio.h>

#include<stdlib.h>

/\* Link list node \*/

struct node

{

int data;

struct node\* next;

};

struct node\* reverse(struct node\* head\_ref, int k)

{

struct node\* prev = NULL;

struct node\* current = head\_ref;

struct node\* next;

int i=0;

while (i<k)

{

next = current->next;

current->next = prev;

prev = current;

current = next;

i++;

}

next=prev;

while(prev->next!=NULL)

prev=prev->next;

prev->next=current;

head\_ref = next;

return head\_ref;

}

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;

}

}

/\* Drier program to test above function\*/

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, 3);

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

printList(head);

return(0);

}