**DS LAB TEST 2**

/\***Given a Doubly linked list, rotate the n number of nodes of the list.**

**Sample Input:**

**n=3**

**1&lt;-&gt;2&lt;-&gt;3&lt;-&gt;4&lt;-&gt;5**

**Sample Output:4&lt;-&gt;5&lt;-&gt;1&lt;-&gt;2&lt;-&gt;3**\*/

#include<stdio.h>

#include<stdlib.h>

struct Node {

int data;

struct Node\* next;

};

void rot(struct Node\*\* head, int t)

{

if (t == 0)

return;

struct Node\* current = \*head;

int count = 1;

while (count < t && current != NULL) {

current = current->next;

count++;

}

if (current == NULL)

return;

struct Node\* nNode = current;

while (current->next != NULL)

current = current->next;

current->next = \*head;

\*head = nNode->next;

nNode->next = NULL;

}

void push(struct Node\*\* nhead, int ndata)

{

struct Node\* newnode = (struct Node\*)malloc(sizeof(struct Node));

newnode->data = ndata;

newnode->next = (\*nhead);

(\*nhead) = newnode;

}

void display(struct Node\* node)

{

while (node != NULL) {

printf("%d ", node->data);

node = node->next;

}

}

int main(void)

{

struct Node\* nhead = NULL;

int d,no;

printf("\nNumber of entries you want: ");

scanf("%d",&no);

for (int i = no; i > 0; i -= 1){

printf("Enter element %d in the list: ",i);

scanf("%d",&d);

push(&nhead, d);

}

printf("\n\t\*\*\*Linked List\*\*\*\n");

display(nhead);

rot(&nhead, 3);

printf("\n\t\*\*\*After Rotating\*\*\*\n");

display(nhead);

return (0);

}

