**DS LAB-6**

Q.)WAP to Implement Singly Linked List with following operations

a) Create a linked list.

b) Insertion of a node at first position, at any position and at end of list.

c) Deletion of first element, specified element and last element in the list.

d) Display the contents of the linked list.

#include<stdio.h>

#include<stdlib.h>

struct node {

int data;

struct node \*next;

};

struct node\* head;

int length() {

struct node\*temp = head;

int cnt = 0;

while(temp->next!=NULL) {

cnt++;

temp = temp->next;

}

return cnt+1;

}

void insert\_at\_end() {

struct node \*newnode, \*temp;

int item;

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

printf("Enter the data : ");

scanf("%d",&item);

newnode->data = item;

if(head == NULL) {

newnode->next = NULL;

head = newnode;

printf("Node Created\n");

}

else{

temp = head;

while(temp->next!=NULL){

temp = temp->next;

}

temp->next = newnode;

newnode->next = NULL;

printf("Node Created at the end.\n");

}

}

void insert\_at\_first() {

if(head == NULL) {

insert\_at\_end();

return;

}

struct node \*newnode;

int ele;

printf("Enter the element to be inserted at first position : ");

scanf("%d",&ele);

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

newnode->data = ele;

newnode->next = head;

head = newnode;

printf("Element inserted at the first position of the list.\n");

}

void insert\_at\_anypos(int pos) {

if(head==NULL) {

insert\_at\_first();

return;

}

if(pos>length()) {

insert\_at\_end();

return;

}

struct node \*newnode ,\*temp;

temp = head;

int ele;

printf("Enter the element to be inserted : ");

scanf("%d",&ele);

int jump = 1;

while(jump<pos-1) {

temp = temp->next;

jump++;

}

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

newnode->data = ele;

newnode->next = temp->next;

temp->next = newnode;

printf("Element inserted at position %d\n",pos);

}

void delete\_at\_first() {

if(head==NULL) {

printf("No elements present\n");

return;

}

struct node\*temp = head;

head = head->next;

free(temp);

printf("Element at first node deleted\n");

}

void delete\_at\_end() {

if(head == NULL) {

printf("No elements present\n");

return;

}

struct node\*temp = head;

while(temp->next->next!=NULL) {

temp = temp->next;

}

free(temp->next);

temp->next = NULL;

printf("Element at last deleted\n");

}

void delete\_at\_anypos(int pos) {

if(head == NULL) {

printf("No elements present\n");

return;

}

if(pos == 1) {

delete\_at\_first();

return;

}

if(pos > length()) {

delete\_at\_end();

return;

}

struct node\*prev = head;

int jump = 1;

while(jump<pos-1) {

prev = prev->next;

jump++;

}

struct node\*temp = prev->next;

prev->next = temp->next;

free(temp);

printf("Element at %d deleted\n",pos);

}

void display() {

struct node\* ptr = NULL;

ptr = head;

if(ptr == NULL)

printf("No data to print\n");

else{

printf("List Contents : \n");

while(ptr!=NULL) {

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

ptr = ptr->next;

}

printf("\n");

}

}

int main() {

int choice,pos;

printf("1.Insert at back\n2.Insert at front\n3.Insert at any position\n4.Delete at front\n5.Delete at end\n6.Delete at any position\n7.Display\n8.Exit\n");

printf("Enter your choice : ");

scanf("%d",&choice);

while(choice!=8) {

if(choice == 1) {

insert\_at\_end();

}

if(choice == 2) {

insert\_at\_first();

}

if(choice == 3) {

printf("Enter postion of the element you want to insert : ");

scanf("%d",&pos);

insert\_at\_anypos(pos);

}

else if(choice == 4) {

delete\_at\_first();

}

else if(choice == 5) {

delete\_at\_end();

}

else if(choice == 6) {

printf("Enter postion of the element you want to delete : ");

scanf("%d",&pos);

delete\_at\_anypos(pos);

}

else if(choice == 7) {

display();

}

printf("Enter your next choice : ");

scanf("%d",&choice);

}

return 0;

}

**OUTPUT SCREEN:**

