1. WAP to Implement Singly Linked List with following operations a) Createalinkedlist. b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linked list.

Program:

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

struct Node

{

int data;

struct Node\* next;

};

struct Node\* createNode(int data)

{

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node)); newNode->data = data;

newNode->next = NULL; return newNode;

}

void insertAtFirst(struct Node\*\* head, int data)

{

struct Node\* newNode = createNode(data); newNode->next = \*head;

\*head = newNode;

}

void insertAtEnd(struct Node\*\* head, int data)

{

struct Node\* newNode = createNode(data); if (\*head == NULL)

{

\*head = newNode; return;

}

struct Node\* temp = \*head; while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = newNode;

}

void insertAtPosition(struct Node\*\* head, int data, int position)

{

struct Node\* newNode = createNode(data); if (position == 0)

{

insertAtFirst(head,data); return;

}

struct Node\* temp = \*head;

for (int i = 0; temp != NULL && i < position - 1; i++)

{

temp = temp->next;

}

if (temp == NULL)

{

printf("Position out of range\n"); free(newNode);

return;

}

newNode->next = temp->next; temp->next = newNode;

}

void display(struct Node\* head)

{

struct Node\* temp = head; while (temp != NULL)

{

printf("%d -> ", temp->data); temp = temp->next;

}

printf("NULL\n");

}

int main()

{

struct Node\* head = NULL;

printf("Linked list after inserting the node:10 at the beginning \n"); insertAtFirst(&head, 10);

display(head);

printf("Linked list after inserting the node:20 at the end \n"); insertAtEnd(&head, 20);

display(head);

printf("Linked list after inserting the node:1 at the end \n"); insertAtPosition(&head,30,1);

display(head);

}

