WAP to Implement Singly Linked List with following operations a) Create a linked list. b) Deletion of first element, specified element and last element in the list. c) Display the contents of the linked list.

Program:

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

struct node { int value;

struct node\* next;

};

typedef struct node\* NODE; NODE get\_node() {

NODE ptr = (NODE)malloc(sizeof(struct node)); if (ptr == NULL) {

printf("Memory not allocated\n");

}

return ptr;

}

NODE delete\_first(NODE first) { NODE temp = first;

if (first == NULL) { printf("Linked list is empty\n"); return NULL;

}

first = first->next; free(temp);

return first;

}

NODE delete\_last(NODE first) { NODE prev, last;

if (first == NULL) { printf("Linked list is empty\n"); return NULL;

}

prev = NULL; last = first;

while (last->next != NULL)

{

prev = last;

last = last->next;

}

if (prev == NULL)

{

free(first); return NULL;

}

prev->next = NULL; free(last);

return first;

}

NODE delete\_value(NODE first, int value\_del) { if (first == NULL) {

printf("Linked list is empty\n"); return NULL;

}

NODE prev = NULL;

NODE current = first;

while (current != NULL && current->value != value\_del) { prev = current;

current = current->next;

}

if (current == NULL) { printf("Value not found\n"); return first;

}

if (prev == NULL) { first = current->next;

} else {

prev->next = current->next;

}

free(current); return first;

}

void display(NODE first) { NODE temp = first;

if (first == NULL) { printf("Empty\n"); return;

}

while (temp != NULL) { printf("%d ", temp->value); temp = temp->next;

}

printf("\n");

}

NODE insert\_beginning(NODE first, int item) { NODE new\_node = get\_node();

new\_node->value = item; new\_node->next = first; return new\_node;

}

int main() {

NODE head = NULL;

int choice, item;

head = insert\_beginning(head, 1); head = insert\_beginning(head, 2); head = insert\_beginning(head, 3); head = insert\_beginning(head, 4);

while (1) {

printf("1. Delete first\n"); printf("2. Delete last\n"); printf("3. Delete value\n"); printf("4. Display\n"); printf("5. Exit\n"); printf("Enter your choice: "); scanf("%d", &choice);

switch (choice) { case 1:

head = delete\_first(head); break;

case 2:

head = delete\_last(head); break;

case 3:

printf("Enter value to delete: "); scanf("%d", &item);

head = delete\_value(head, item); break;

case 4:

display(head); break;

case 5:

return 0; default:

printf("Invalid choice\n");

}

}

return 0;

}

