**Write a menu-driven C program to perform basic operations on Linear Singly Linked List.**

**#include <stdio.h>**

**#include <conio.h>**

**#include <stdlib.h>**

**struct linked\_list**

**{**

**int data;**

**struct linked\_list \* next;**

**};**

**void main()**

**{**

**struct linked\_list \*newnode = NULL, \*startnode = NULL, \*endnode = NULL, \*node = NULL, \*delnode = NULL, \*prevnode = NULL;**

**int value = 0, choice = 0, loc = 0, i = 0;**

**clrscr();**

**do**

**{**

**clrscr();**

**printf("\n 1. APPEND or INSERT END");**

**printf("\n 2. DISPLAY");**

**printf("\n 3. INSERT BEGINNING");**

**printf("\n 4. DELETE");**

**printf("\n 5. INSERT BEFORE");**

**printf("\n 6. INSERT AFTER");**

**printf("\n 7. EXIT");**

**printf("\n Enter your choice: ");**

**scanf("%d", &choice);**

**switch(choice)**

**{**

**case 1:**

**printf("\n Enter a value to insert: ");**

**scanf("%d", &value);**

**newnode = (struct linked\_list \*) malloc(sizeof(struct linked\_list));**

**newnode -> data = value;**

**newnode -> next = NULL;**

**if(startnode == NULL)**

**{**

**startnode = newnode;**

**}**

**else**

**{**

**endnode -> next = newnode;**

**}**

**endnode = newnode;**

**break;**

**case 2:**

**if(startnode == NULL)**

**{**

**printf("\n List is empty.");**

**}**

**else**

**{**

**for(node=startnode; node != NULL; node=node->next)**

**{**

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

**}**

**}**

**break;**

**case 3:**

**printf("\n Enter a value to insert: ");**

**scanf("%d", &value);**

**newnode = (struct linked\_list \*) malloc(sizeof(struct linked\_list));**

**newnode -> data = value;**

**newnode -> next = startnode;**

**startnode = newnode;**

**break;**

**case 4:**

**printf("\n Enter a value to delete: ");**

**scanf("%d", &value);**

**for(node=startnode; node != NULL; node=node->next)**

**{**

**if(node -> data == value)**

**{**

**delnode = node;**

**if(node == startnode)**

**{**

**startnode = startnode -> next;**

**}**

**else if(node == endnode)**

**{**

**prevnode -> next = NULL;**

**endnode = prevnode;**

**}**

**else**

**{**

**prevnode ->next = node -> next;**

**}**

**free(delnode);**

**break;**

**}**

**prevnode = node;**

**}**

**break;**

**case 5:**

**printf("\n Enter a location: ");**

**scanf("%d", &loc);**

**printf("\n Enter a value to insert: ");**

**scanf("%d", &value);**

**node=startnode;**

**for(i = 1; i < loc; i++)**

**{**

**prevnode = node;**

**node = node -> next;**

**}**

**newnode = (struct linked\_list \*) malloc(sizeof(struct linked\_list));**

**newnode -> data = value;**

**if(node == startnode)**

**{**

**newnode -> next = startnode;**

**startnode = newnode;**

**}**

**else**

**{**

**prevnode -> next = newnode;**

**newnode -> next = node;**

**}**

**break;**

**case 6:**

**printf("\n Enter a location: ");**

**scanf("%d", &loc);**

**printf("\n Enter a value to insert: ");**

**scanf("%d", &value);**

**node=startnode;**

**for(i = 1; i <= loc; i++)**

**{**

**prevnode = node;**

**node = node -> next;**

**}**

**newnode = (struct linked\_list \*) malloc(sizeof(struct linked\_list));**

**newnode -> data = value;**

**if(node == endnode)**

**{**

**newnode -> next = NULL;**

**endnode -> next = newnode;**

**endnode = newnode;**

**}**

**else**

**{**

**prevnode -> next = newnode;**

**newnode -> next = node;**

**}**

**break;**

**case 7:**

**exit(0);**

**break;**

**default:**

**printf("\n Invalud choice");**

**}**

**getch();**

**} while(choice != 7);**

**}**