**DATA STRUCTURES LAB**

**PROGRAM 7**

**Develop a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Programme, Sem, PhNo**

**a. Create a SLL of N Students Data by using *front insertion*.**

**b. Display the status of SLL and count the number of nodes in it**

**c. Perform Insertion / Deletion at End of SLL**

**d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack)**

**e. Exit**

Program:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct stud {

char usn[11], name[15], branch[4], phno[11];

int sem;

struct stud \*next;

} \*f = NULL, \*r = NULL, \*t = NULL;

void ins(int ch) {

t = (struct stud\*)malloc(sizeof(struct stud));

printf("\nEnter USN: ");

scanf("%s", t->usn);

printf("Enter Name: ");

scanf("%s", t->name);

printf("Enter Branch: ");

scanf("%s", t->branch);

printf("Enter Sem: ");

scanf("%d", &t->sem);

printf("Enter Phno: ");

scanf("%s", t->phno);

t->next = NULL;

if (!r) { // If list is empty

f = r = t;

} else {

if (ch) { // Insert at end

r->next = t;

r = t;

} else { // Insert at beginning

t->next = f;

f = t;

}

}

}

void del(int ch) {

if (!f) {

printf("\nList Empty");

} else {

struct stud \*t1;

// Case when only one element is in the list

if (f == r) {

t1 = f;

f = r = NULL;

} else if (ch) { // Delete from the rear

t1 = r;

struct stud \*temp;

for (temp = f; temp->next != r; temp = temp->next);

r = temp;

r->next = NULL;

} else { // Delete from the front

t1 = f;

f = f->next;

}

// Display deleted element's details

printf("\nElement deleted is:\n");

printf("USN: %s\nName: %s\nBranch: %s\nSem: %d\nPhno: %s\n", t1->usn, t1->name, t1->branch, t1->sem, t1->phno);

// Free the memory of the deleted node

free(t1);

}

}

void disp() {

if (!f) {

printf("\nList Empty!!!");

} else {

printf("\nList elements are:\n");

for (t = f; t; t = t->next) {

printf("\nUSN: %s\nName: %s\nBranch: %s\nSem: %d\nPhno: %s\n", t->usn, t->name, t->branch, t->sem, t->phno);

}

}

}

int main() {

int ch, n, i;

printf("\n........Menu..........,\n");

printf("1. Create\n");

printf("2. Display\n");

printf("3. Insert at end\n");

printf("4. Delete at end\n");

printf("5. Insert at beginning\n");

printf("6. Delete at beginning\n");

printf("7. Exit\n");

while (1) {

printf("\nEnter choice: ");

scanf("%d", &ch);

switch (ch) {

case 1:

printf("\nEnter no. of nodes: ");

scanf("%d", &n);

for (i = 0; i < n; i++)

ins(0);

break;

case 2:

disp();

break;

case 3:

ins(1);

break;

case 4:

del(1);

break;

case 5:

ins(0);

break;

case 6:

del(0);

break;

case 7:

exit(0);

default:

printf("\nInvalid choice!!!!");

}

}

return 0;

}