**DATA STRUCTURES LAB**

**PROGRAM 8**

**Develop a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo**

**a. Create a DLL of N Employees Data by using *end insertion*.**

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

**c. Perform Insertion and Deletion at End of DLL**

**d. Perform Insertion and Deletion at Front of DLL**

**e. Demonstrate how this DLL can be used as Double Ended Queue.**

**f. Exit**

Program:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int count = 0;

struct node {

struct node \*prev;

int ssn, phno;

float sal;

char name[20], dept[10], desg[20];

struct node \*next;

} \*h = NULL, \*temp = NULL, \*temp1 = NULL;

void create() {

int ssn, phno;

float sal;

char name[20], dept[10], desg[20];

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

temp->prev = NULL;

temp->next = NULL;

printf("\nEnter SSN, Name, Department, Designation, Salary, and Phone Number of employee: ");

scanf("%d %s %s %s %f %d", &ssn, name, dept, desg, &sal, &phno);

temp->ssn = ssn;

strcpy(temp->name, name);

strcpy(temp->dept, dept);

strcpy(temp->desg, desg);

temp->sal = sal;

temp->phno = phno;

count++;

}

void insertbeg() {

create();

if (h == NULL) {

h = temp;

temp1 = h;

} else {

temp->next = h;

h->prev = temp;

h = temp;

}

}

void insertend() {

create();

if (h == NULL) {

h = temp;

temp1 = h;

} else {

temp1->next = temp;

temp->prev = temp1;

temp1 = temp;

}

}

void displaybeg() {

struct node \*temp2 = h;

if (temp2 == NULL) {

printf("List is empty to display.\n");

return;

}

printf("\nLinked list elements from beginning:\n");

while (temp2 != NULL) {

printf("%d %s %s %s %.2f %d\n", temp2->ssn, temp2->name, temp2->dept, temp2->desg, temp2->sal, temp2->phno);

temp2 = temp2->next;

}

printf("Number of employees = %d\n", count);

}

int deleteend() {

if (h == NULL) {

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

return 0;

}

if (h->next == NULL) { // Only one element

printf("%d %s %s %s %.2f %d\n", h->ssn, h->name, h->dept, h->desg, h->sal, h->phno);

free(h);

h = NULL;

temp1 = NULL;

} else {

struct node \*temp2 = temp1->prev;

temp2->next = NULL;

printf("%d %s %s %s %.2f %d\n", temp1->ssn, temp1->name, temp1->dept, temp1->desg, temp1->sal, temp1->phno);

free(temp1);

temp1 = temp2;

}

count--;

return 0;

}

int deletebeg() {

if (h == NULL) {

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

return 0;

}

struct node \*temp = h;

if (h->next == NULL) { // Only one element

printf("%d %s %s %s %.2f %d\n", h->ssn, h->name, h->dept, h->desg, h->sal, h->phno);

free(h);

h = NULL;

temp1 = NULL;

} else {

h = h->next;

h->prev = NULL;

printf("%d %s %s %s %.2f %d\n", temp->ssn, temp->name, temp->dept, temp->desg, temp->sal, temp->phno);

free(temp);

}

count--;

return 0;

}

int main() {

int ch, n, i;

printf("-----------------MENU--------------------\n");

printf("1 - Create a DLL of n employees\n");

printf("2 - Display from beginning\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");

printf("------------------------------------------\n");

while (1) {

printf("\nEnter choice: ");

scanf("%d", &ch);

switch (ch) {

case 1:

printf("\nEnter number of employees: ");

scanf("%d", &n);

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

insertend();

}

break;

case 2:

displaybeg();

break;

case 3:

insertend();

break;

case 4:

deleteend();

break;

case 5:

insertbeg();

break;

case 6:

deletebeg();

break;

case 7:

exit(0);

default:

printf("Wrong choice!\n");

}

}

return 0;

}