#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Node {

int DeptID;

char DeptName[50];

int NoEmpDept;

struct Node\* prev;

struct Node\* next;

} Node;

// Function to create a new node

Node\* createNode(int DeptID, char\* DeptName, int NoEmpDept) {

Node\* newNode = (Node\*)malloc(sizeof(Node));

newNode->DeptID = DeptID;

strcpy(newNode->DeptName, DeptName);

newNode->NoEmpDept = NoEmpDept;

newNode->prev = NULL;

newNode->next = NULL;

return newNode;

}

// Function to insert a node at the end of the DLL

void insertEnd(Node\*\* head, int DeptID, char\* DeptName, int NoEmpDept) {

Node\* newNode = createNode(DeptID, DeptName, NoEmpDept);

if (\*head == NULL) {

\*head = newNode;

} else {

Node\* temp = \*head;

while (temp->next != NULL) {

temp = temp->next;

}

temp->next = newNode;

newNode->prev = temp;

}

printf("Department added successfully at the end.\n");

}

// Function to display the DLL and count the number of departments

void display(Node\* head) {

if (head == NULL) {

printf("The DLL is empty.\n");

return;

}

int count = 0;

Node\* temp = head;

printf("Doubly Linked List of Departments:\n");

while (temp != NULL) {

printf("DeptID: %d, DeptName: %s, NoEmpDept: %d\n", temp->DeptID, temp->DeptName, temp->NoEmpDept);

temp = temp->next;

count++;

}

printf("Total number of departments: %d\n", count);

}

// Function to insert a node at the front of the DLL

void insertFront(Node\*\* head, int DeptID, char\* DeptName, int NoEmpDept) {

Node\* newNode = createNode(DeptID, DeptName, NoEmpDept);

if (\*head == NULL) {

\*head = newNode;

} else {

newNode->next = \*head;

(\*head)->prev = newNode;

\*head = newNode;

}

printf("Department added successfully at the front.\n");

}

// Function to delete a node from the end of the DLL

void deleteEnd(Node\*\* head) {

if (\*head == NULL) {

printf("The DLL is empty, nothing to delete.\n");

return;

}

Node\* temp = \*head;

if (temp->next == NULL) { // Only one node

free(temp);

\*head = NULL;

} else {

while (temp->next != NULL) {

temp = temp->next;

}

temp->prev->next = NULL;

free(temp);

}

printf("Department deleted successfully from the end.\n");

}

// Function to delete a node from the front of the DLL

void deleteFront(Node\*\* head) {

if (\*head == NULL) {

printf("The DLL is empty, nothing to delete.\n");

return;

}

Node\* temp = \*head;

if (temp->next == NULL) { // Only one node

free(temp);

\*head = NULL;

} else {

\*head = temp->next;

(\*head)->prev = NULL;

free(temp);

}

printf("Department deleted successfully from the front.\n");

}

// Main function to drive the menu

int main() {

Node\* head = NULL;

int choice, DeptID, NoEmpDept;

char DeptName[50];

do {

printf("\n--- College Departments Menu ---\n");

printf("1. Create DLL (Insert End)\n");

printf("2. Display DLL and Count Departments\n");

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

printf("4. Delete from End\n");

printf("5. Delete from Front\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter DeptID: ");

scanf("%d", &DeptID);

printf("Enter DeptName: ");

scanf(" %[^\n]s", DeptName); // Read string with spaces

printf("Enter NoEmpDept: ");

scanf("%d", &NoEmpDept);

insertEnd(&head, DeptID, DeptName, NoEmpDept);

break;

case 2:

display(head);

break;

case 3:

printf("Enter DeptID: ");

scanf("%d", &DeptID);

printf("Enter DeptName: ");

scanf(" %[^\n]s", DeptName);

printf("Enter NoEmpDept: ");

scanf("%d", &NoEmpDept);

insertFront(&head, DeptID, DeptName, NoEmpDept);

break;

case 4:

deleteEnd(&head);

break;

case 5:

deleteFront(&head);

break;

case 6:

printf("Exiting the program.\n");

break;

default:

printf("Invalid choice. Please try again.\n");

}

} while (choice != 6);

return 0;

}