6) Singly Linked List Deletion Program

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void create();
void del(char*);
void display();
void insertbeg();
struct node *delete_beg(struct node *);
struct node *delete_end(struct node *);
struct node {
char name[20];
char id[10];
int sem;
struct node * next;
};
struct node* head = NULL;
void main() {
int c;
char ele[10];
do {
printf("Enter Choice :\n1. Create \n2. Display \n3. Delete a specific ID \n
4. Insert in the beginning \n5. Delete a node from the beginning \n6. Delete
a node from the end \n7. Exit \n");
scanf("%d", &c);
switch(c) {
case 1 : create(); break;
case 2 : display(); break;
case 3 : printf("Enter the element id to be deleted \n");
scanf("%s", ele);
```

```
del(ele); break;
case 4: insertbeg(); break;
case 5: head = delete_beg(head); break;
case 6: head = delete_end(head); break;
case 7: exit(0); break;
}
}while(1);
}
void create(){
struct node *newnode, *temp;
char n[20], id1[10];
int s;
newnode = (struct node *) malloc(sizeof(struct node));
printf("Enter the Name, USN, Semester: \n");
scanf("%s", n);
scanf("%s", id1);
scanf("%d", &s);
strcpy(newnode->name, n);
strcpy(newnode->id, id1);
newnode->sem = s;
if(head == NULL){
newnode->next = NULL;
head = newnode;
printf("Node is created\n");
}else{
temp = head;
while(temp->next != NULL)
temp = temp->next;
temp->next = newnode;
newnode->next = NULL;
printf("NODE created\n");
}
```

```
}
void display () {
struct node *ptr = NULL;
ptr = head;
if (ptr == NULL) {
printf ("No element to print \n");
} else {
while (ptr != NULL) {
puts (ptr -> name);
puts (ptr -> id);
printf ("%d \n", ptr -> sem);
ptr = ptr -> next;
}
}
}
void del (char id1[10]) {
struct node *temp, *del = NULL;
if (head == NULL) {
printf ("Empty list \n"); return;
}
temp = head;
if (strcmp (head -> id, id1) == 0) {
head = head -> next;
return;
}
while (temp -> next != NULL) {
if ((temp -> next -> id, id1) == 0) {
del = temp -> next;
if (del -> next == NULL)
temp -> next = NULL;
else
temp -> next = del -> next;
```

```
}
else
temp -> next = del -> next ;
}
if (del == NULL)
printf("Element not found \n"); return ;
}
void insertbeg() {
struct node *newnode ;
char n[20], id1[10] ;
int s ;
printf("Enter the elements : Name, USN and Sem \n");
scanf("%s", n) ;
scanf("%s", id1) ;
scanf("%d", &s);
newnode = (struct node*) malloc (sizeof(struct node)) ;
strcpy (newnode -> name, n) ;
strcpy (newnode -> id, id1) ;
scanf("%d", &s) ;
newnode -> next = head ;
head = newnode ;
}
struct node *delete_beg (struct node *head) {
struct node *ptr ;
ptr = head ;
head = head -> next ;
free (ptr) ;
return head ;
}
struct node *delete_end (struct node *head) {
struct node *ptr, *preptr ;
ptr = head ;
```

```
while (ptr->next != NULL) {
preptr = ptr;
ptr = ptr->next;
}
preptr->next = NULL;
free(ptr);
return head;
}
```