IBM19CS006

27/11/20 Lab-6 Deletion in list

```
#include<stdio.h>
#include<stlib.h>
void create();
void delete_begin();
void delete_end();
void delete_pos();
void display();
struct node
{  int id;
   char name[100];
   struct node *next;
};
struct node *head=NULL;
int main()
{  int choice;
   do{
   printf("1.Create \n2. Display \n 3. Delete from
beginning \n4. Delete from end \n5. Delete from
position \n6. Exit \n");
printf("Enter your choice: \t");
scanf("%d", &choice);
switch(choice)
{

case 1: create();
        break;
case 2: display();
        break;
case 3: delete_begin();
```

```
        break;
case 4: delete_end();
        break;
case 5: delete_pos();
        break;
case 6: exit(0);
        break;
        }
}while(choice!=6);
return 0;
}

void create(){
struct node *temp, *ptr;
temp=(struct node*)malloc(sizeof(struct node));
printf("\n Enter student name: \n");
scanf("%s", temp->name);
printf("\n Enter student id: \n");
scanf("%d", &temp->id);
temp->next=NULL;
if(head==NULL){
    head=temp;
}
else{
    ptr=head;
while(ptr->next!=NULL)
{   ptr=ptr->next;
}
ptr->next=temp;
```

```
    temp -> next = NULL;
    }
  }
}

void display()
{ struct node *ptr;
  if (head == NULL)
  {
    printf("\n List is empty \n");
    return;
  }
  else {
    ptr = head;
    printf("The list is:\n");
    while (ptr != NULL)
{
    printf("Student name: %s \n", ptr -> name);
    printf("Student ID: %d \n", ptr -> id);
     ptr = ptr -> next;
  }
}
}

void delete_begin(){
struct node *ptr;
if (head == NULL)
{
printf("\n List is empty \n");
return;
}
```

```
else {
    ptr = head;
    head = head -> next;
printf ("\n Deleted node: %d \t", ptr->id);
    free (ptr);
    }
    }
    void delete_end()
{   struct node *temp, *ptr;
    if (head == NULL)
    {
    printf ("\n List is empty \n");
    exit (0);
    }
    else if (head -> next == NULL)
    {   ptr = head;
        head = NULL;
        printf ("\n Deleted node: %d \t", ptr->id);
```