1BM19CS005
Adithi Girimaji

classmate
Date
Page

Lab-5 program:

```
#include<stdio.h>
# include<stdlib.h>
void create();
void display();
void insert_pos1();
void insert_bef();
void insert_aft();
void insert_last();
void insert_pos();
struct node
{   int id;
    char name[50];
    int sem;
    struct node *next;
};
struct node *head = NULL;
int main(int argc, char **argv)
{
   int choice, ch;
   do
   {
   printf("1. Create\n2. Display\n3. Insert at position1\n4. Insert before an element\n5. Insert after an element\n6. Insert at the end of the list\n7. Insert at any position mentioned\n");
   printf("\n Enter your choice: ");
   scanf("%d", &choice);
   switch(choice)
   {
```

```
case 1: create(); break;
case 2: display(); break;
case 3: insert_pos(); break;
case 4: insert_bef(); break;
case 5: insert_aft(); break;
case 6: insert_last(); break;
case 7: insert_pos(); break;
default: printf("Wrong choice");
}
printf("\n Press 1 if you want to continue else a
other number\n");
scanf("%d", &ch);
} while(ch==1);
return 0;
}
void create()
{ struct node *newnode, *temp;
int ID, s, j;
char n[50];
newnode = (struct node*) malloc (sizeof(struct node));
printf("Enter Student id, name and sem: ");
scanf("%d", &ID);
scanf("%s", n);
scanf("%d", &s);
newnode->id = ID;
for(j=0;; j++)
{ newnode->name[j]=n[j];
if(n[j]=='\0') break; }
newnode->sem = s;
if(head == NULL)
{ newnode->next = NULL;
head = newnode;
```


Scanned with CamScanner

```
        printf("Node is created\n");
    } else {
        temp=head;
        while(temp->next!=NULL)
        {   temp=temp->next;
        } temp->next=newnode;
        newnode->next=NULL;
        printf("Node is created\n");
    }
void display()
{   struct node *ptr=NULL;
    ptr=head;
    if(ptr==NULL)
    {printf("Nothing to print\n");}
    else {
    while(ptr!=NULL){
    printf("\tID:%d\tName:%s\tSem:%d\n",ptr->id,
                    ptr->name,ptr->sem);
    ptr=ptr->next;
    }}}
void insertpos1()
{   struct node *newnode;
    int ID,s,j; char n[50];
    printf("Enter student id name and sem");
    scanf("%d",&ID);
    scanf("%s",n);
    scanf("%d",&s);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->id=ID;
    for(j=0;;j++)
    {   newnode->name[j]=n[j];
        if(n[j]=='\0') break;  }
```


Scanned with CamScanner

```
    newnode->sem=s;
    newnode->next=head;
    head=newnode;
}
void insert_bef()
{  struct node *newnode, *temp1, *temp2=NULL;
   temp1=head;
   int ID, s, ele_bef, j; char n[50];
   printf("Enter the StudentID before which the d
   has to be inserted: ");
   scanf("%d", &ID);
   scanf("%s", n);
   scanf("%d", &s);
   newnode=(struct node*)malloc(sizeof(struct
   newnode->id=ID;
   for(j=0;;j++)
   { newnode->name[j];
     if(n[j]=='\0') break; }
   newnode->sem=s;
   if(head->id==ele_bef)
   { newnode->next=head;
     head=newnode; return;
   } while(temp1->next!=NULL)
   { if(temp1->next->id==ele_bef)
     { temp2=temp1->next;
       temp1->next=newnode;
       newnode->next=temp2; return;
     } else temp1=temp1->next; }
   if(temp2==NULL)
     printf("Element is not found in the list\n");
     return;
} }
```


Scanned with CamScanner

```
void insert_aft()
{ struct node *newnode, *temp1, *temp2 = NULL;
  temp1 = head;
  int ID, ele_aft, s, j;
  char n[50];
  printf("Enter student ID after which element has to
  be inserted : ");
  scanf("%d", &ele_aft);
  printf("Enter the student id, name, and sem:");
  scanf("%d", &ID);
  scanf("%s", n);
  scanf("%d", &s);
  newnode = (struct node*)malloc(sizeof(struct node));
  newnode -> id = ID;
for(j=0; j++)
  { newnode -> name[j] = n[j];
    if(n[j] == '\0') break; }
  newnode -> sem = s;
  while (temp1 -> next != NULL)
  { if (temp1 -> id == ele_aft)
      { temp2 = temp1 -> next;
        temp1 -> next = newnode;
        newnode -> next = temp2; return;
      } else temp1 = temp1 -> next; }
  if (temp1 -> id == ele_aft)
  { if (temp -> id == ele_aft)
    { temp1 -> next = newnode;
      newnode -> next = NULL;
    } return;
  if (temp2 == NULL)
    { printf ("Element is not found in the list\n");
      return; } }
```


Scanned with CamScanner

```
void insert_last()
{ struct node *newnode, *ptr;
  ptr=head; int ID, s, j; char n[50];
  printf(" Enter student id, name and sem: ");
   scanf ("%d ", &ID);
   scanf ("%s", n);
   scanf ("%d", &s);
  newnode = (struct node*) malloc (sizeof (struct node));
  newnode -> id = ID;
  for(j=0;; j++)
   { newnode -> name[j] = n[j];
     if (n[j] == '\0') break; }
   newnode -> sem = s;
   do { if (ptr -> next == NULL)
        { ptr -> next = newnode;
          newnode -> next = NULL; return; }
      ptr = ptr -> next;
    }while(1); }
void insert_pos()
{  int pos, c=1, j, ID, s; char n[50]; struct node *temp;
   temp = head;
   if (temp != NULL && temp -> next != NULL)
   { do {
        if (temp -> next != NULL)
        { temp = temp -> next; c++; }
      if (temp -> next == NULL)
      break;
   } while (1);
   }
   printf ("Enter the position at which the element has
     to be inserted \n");
     scanf ("%d", &pos);
```


Scanned with CamScanner

```
if (pos<1||pos>ct+1)
{ printf("Element cannot be inserted\n");
    return;
}
struct node *newnode;
newnode = (struct node*) malloc (sizeof (struct node));
printf ("Enter the student id, name, sem:");
scanf ("%d", &ID);
scanf ("%s", n);
scanf ("%d", &s);
newnode -> id= ID;
for (j=0;; j++)
{ newnode -> name[j] = n[j];
    if (n[j] == '\0') break;
}
newnode-> sem = s;
if (pos==1)
  { newnode -> next = head;
    head = newnode; }
else if (pos == ct+1)
{   temp = head;
    do { if (temp->next == NULL)
        { temp-> next= newnode;
          newnode -> next = NULL; return;
        }
      temp = temp -> next;
  } while (1);
} else { struct node *temp2; temp= head;
    for (int i= 1; i< pos-1; i++)
      { temp = temp-> next; }
    temp2 = temp ->next;
    temp->next = newnode; newnode -> next = temp2;
}
```


Scanned with CamScanner