Date:

LAB 6

Page No.

```
#include <Stdio.h>
#include <Stdlib.h>
#include <string.h>
struct node
{ int sem;
   Char name[30];
   char ID[30];
   struct node *next;
};
struct node *addatbeg(struct node *start, int sem,
char n[], char I[])
{
  struct node *tmp;
  tmp = (struct node *) malloc (sizeof(struct node));
  tmp->sem = sem;
  memcpy(tmp->name, n, 20);
  memcpy (tmp->ID, I, 20);
  tmp->next = Start;
  start = tmp;
  return start;
}
struct node *create_list (struct node *start)
{  start = NULL;
  fflush (stdin);
  printf ("\n Enter the student ID:\n");
```

Date:

Page No.

```
Char I[30];
scanf("%s", I);
printf("Enter the student name: \n");
char n[30];
Scanf("%s", n);
printf("Enter the semester the student is in: ");
int sem = 0;
Scanf("%d", &sem);
fflush(stdin);
start = addatbeg (start, sem, n , I);
return start;
}
void display (struct node *start)
{   struct node *p;
  if(start == NULL)
{ printf ("List is empty \n");
return;
} p = start;
 printf ("List is :\n");
 while (p! = NULL)
{ print f ("Name : %S\nID : %S\n sem: %d\n",
p->name, p->ID, p->sem);
   p = p->next; }
printf ("\n\n");
}
```

Date:

Page No.

```
struct node *addatend( struct node *start, int sem,
char n[], char I[])
{
 struct node *p, *tmp;
tmp = (struct node *)malloc(size of(struct node));
tmp -> sem = sem;
memcpy(tmp -> name, n, 20);
memcpy (tmp -> ID, I, 20);
p = start;
while (p -> next != NULL)
 p = p -> next;
 p -> next = tmp;
 tmp -> next = NULL;
 return start;
}
struct node *addatpos(struct node *start, int sem,
int pos, char n[], char I[])
{ struct node *tmp, *p;
int i;
tmp = (struct node *)malloc(sizeof(struct node));
tmp -> sem = sem;
memcpy(tmp -> name, n, 20);
memcpy(tmp -> ID, I, 20);
 if (pos == 1)
{
```

```
tmp->next =start;
start =tmp;
return start;
}
p=start;
for(i=1; i<pos-1 && p!=NULL; i++)
p=p->next;
if(p==NULL)
printf("There are less than %d elements\n", pos);
else{
tmp ->next =p->next;
p->next=tmp;
} return start;
} struct node *del (struct node *start, char I[])
{ struct node *tmp, *p;
if (start == NULL)
{ printf ("List is empty \n");
return start;
} if (strcmp(start ->ID, I) ==0)
{ tmp= start;
start = start -> next;
free(tmp);
return start;
}
p= start;
```

Date:

Page No.

```
tmp = start;
while (p->next! =NULL)
{ if (strcmp (p->ID, I)==0)
 {  printf ("\n Deleted element!\n");
 tmp -> next =p->next;
return start; }
tmp = p;
p=p->next;
}
if (tmp-next==NULL && strcmp(p->ID, I)==0)
{  tmp->next =NULL;
printf ("\n Delted element !\n");
return start; }
printf ("ID %S not found.\n", I);
return start; }
struct node *delatend (struct node *start)
{     struct node *temp;
 if (start ==NULL)
{printf ("\n List in empty!\n");
return start; }
if (start -> next == NULL)
{ printf ("\n Start deleted\n");
 return NULL;
} temp=start;
while (temp ->next != NULL && temp->next->next!=
NULL)
```

Date:

Page No.

```
{ temp = temp->next;
} temp -> next = NULL;
printf ("\n Deleted at end! \n");
return start;
} struct node *delatbeg (struct node *start)
{ if (start ==NULL)
{ printf (" \n List is empty! \n");
return start; }
if (start ->next ==NULL)
{    start ->next =NULL;
printf ("\n Start deleted \n");
return start; }
else
{    start = start ->next;
   printf ("\n Start deleted \n");
   return   start; }
}
struct node *delatpos (struct node *start, int pos)
{ struct node *p;
  int i;
p= (struct node *)malloc (sizeof (struct node));
if (pos==0)
{ if (start ==NULL)
    return NULL; else
 return start->next;
}
```

Date:

```
p = start;
for(i=0; i<pos && p!=NULL; i++)
  p=p->next;
 if (p==NULL)
printf("There are less than %d elements\n", pos);
else {
 if (p->next !=NULL)
 p->next=p->next->next;
} return start;
  }
int main() {
struct node *start=NULL;
int choice, sem, pos;
char n[30];
char I [30];
while(1)
{ printf("1 to create List \n");
 printf("2 to Display \n");
 printf("3 to Add to empty List \n");
 printf("4 to Add at end \n");
 printf("5 to Add at position\n");
 printf("6 to delete at end \n");
 printf("7 to delete at beginning \n");
printf("8 to delete a particular ID\n");
 printf("Enter your choice :");
 scanf("%d",&choice);
```

```
switch (choice)
{ case 1: Start = Create _list (start);
          break;
  Case 2:   display (start)
         break;
  Case 3: fflush (stdin);
        printf ("\n Enter the Student ID : \n");
        Scanf ("%S", I);
        printf ("Enter the student name: \n");
        scanf ("%S", n);
        printf ("Enter the semester : \n");
        scanf ("%d" &sem);
        fflush (stdin);
        start = addatbeg (start, sem, n, I);
        break;
  Case 4: fflush (stdin);
        printf ("\n Enter the student ID: \n");
        scanf ("%S" I);
        printf ("Enter the student name: \n");
        scanf ("%S", n);
        printf ("Enter the semester: \n");
        Scanf ("%d", &sem);
        fflush (stdin);
        start = addatend (start, sem, n, I);
        break;
```

```
Case 5: fflush(stdin);
        printf("\n Enter the student ID: \n");
        scanf("%s", I);
        printf("Enter the student name: \n");
        scanf("%s", n);
        printf("Enter the semester: \n");
        scanf("%d", &sem);
        fflush(stdin);
       start= addatend(start, sem, n, I);
       break;
```

Page No.

```
printf("Enter the position at which to insert:");
scanf("%d", &pos);
 start = addatpos(start, sem, pos, n, I);
 break;
Case 6:     start = delatend(start);
        break;
Case 7:  start = delatbeg(start);
         break;
Case 8:  fflush(stdin);
        printf("Enter the ID to be deleted :\n");
        scanf("%s", I);
        start = del(start, I);
        break;
Case 0;
       exit(1);
default: printf("Wrong choice \n");
} }
  return 0; }
```