20/11/20 IBM19CS006

Lab 5 – Singly Linked List

Linked list to display student name, id & sem.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void create();
void insert_before();
void insert_end();
void insert_pos();
void display();

struct st
{
  char name[100];
  int id;
  int sem;
  struct st *next;
};
struct st *head=NULL;

int main()
{ int choice;
  do {
  printf("1. Create List \n");
  printf("2. Insert at the beginning \n");
  printf("3. Insert at the end \n");
  printf("4. Insert at position \n");
  printf("5. Display \n");
```

```
printf ("6. Exit \n");
printf ("Enter your choice:\n");
scanf ("%d", &choice);
switch (choice)
{
case 1: create();
    break;
case 2: insert_before();
    break;
case 3: insert_end();
    break;
case 4: insert_pos();
    break;
case 5: display();
    break;
case 6: exit(0);
    break;
}
} while (choice != 6);
return 0;
}

void create()
{
    struct st *newnode, *temp;
    newnode = (struct st*) malloc (sizeof(struct st));
    printf ("Enter name of student: \n");
    scanf ("%s", newnode -> name);
```

```
printf("Enter student id : \n");
scanf("%d",&newnode->id);
printf("Enter student sem : \n");
scanf("%d",&newnode->sem);
newnode->next = NULL;

if(head==NULL){
    head = newnode;
}
else{
    temp = head;
while(temp->next!=NULL)
{
    temp = temp->next;
}
    temp->next = newnode;
}
}


void insert_before()
{
    struct st *newnode, *temp;
    newnode = (struct st*)malloc(sizeof(struct st));
    if(newnode==NULL)
    {
        printf("\n Insufficient memory \n");
        return;
    }
```

```
printf("Enter the name of the student : \n");
scanf("%s", newnode->name);
printf("Enter the student id : \n");
scanf("%d", newnode->id);
printf("Enter student sem : \n");
scanf("%d", newnode->sem);
newnode->next = NULL;

if(head==NULL){
    head = temp;
    }
else {
    newnode->next = head;
    head = newnode;
    }
}

insert_end void insert_end()
{
struct st* temp, *ptr;
temp = (struct st*) malloc(sizeof(struct st));
printf("Enter the name of the student : \n");
scanf("%s", temp->name);
printf("Enter student id : \n");
scanf("%d", &temp->id);
printf("Enter student sem : \n");
scanf("%d", &temp->sem);
temp->next = NULL;
```

```
if(head==NULL)
  {
    head = temp;
  }
else {
    ptr = head;
   while (ptr -> next != NULL)
    {
      ptr = ptr -> next;
    }
   ptr -> next = temp;
  }
}

void insert_pos()
{
  struct st* ptr, *temp;
  int i, pos;
  temp = (struct st*) malloc(sizeof(struct st));
  printf("Enter the position for the node to
    be inserted: \n");
  scanf("%d", &pos);
  printf("Enter name of the student: \n");
  scanf("%s", temp -> name);
  printf("Enter student id: \n");
  scanf("%d", &temp -> id);
  printf("Enter student sem: \n");
  scanf("%d", &temp -> sem);
```

```
temp->next=NULL;
if (pos==0){
    temp->next=head;
    head=temp;
}
else {
for(i=0;ptr=head; i<pos-1; i++){
    ptr=ptr->next;
    if(ptr==NULL)
    {
        printf("\n Position not found \n");
        return;
    }
}
temp->next=ptr->next;
    ptr->next=temp;
}
}
void display()
{ struct st *temp;
temp=head;
while(temp)
{
printf("\n Student name: %s \n", temp->name);
printf("Std Student Id: %d \n", temp->id);
printf("Student semester: %d \n", temp->sem);
temp=temp->next;
}
}
```