Medha

papergrid

Date: / /

Lab program 6

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    int data;
    struct node *next;
};
void insertlast (struct node **headptr) {
    struct node *newnode, *temp;
    int value;
    printf ("Enter value:");
    scanf ("%d", &value");
    newnode = (struct node *) malloc (sizeof (struct node));
    newnode -> data = value;
    newnode -> next = NULL;
    temp = (*headptr);
    if (*headptr == NULL)
        (*headptr) = newnode;
    else {
        while (temp -> next != NULL)
            temp = temp -> next;
        temp -> next = newnode;
    }
}
void deletefirst (struct node **headptr) {
    struct n
    if (*headptr == NULL)
        printf ("The list is empty \n");
    else if ((*headptr) -> next == NULL)
        (*headptr) = NULL;
    else {
        (*headptr) = (*headptr) -> next;
    }
}
```

```
void deletepos (struct node **headptr) {
  struct node *temp;
  int count = 0, currpos = 1, pos;
  if ((*headptr) == NULL) {
      printf ("List is empty\n");
      return;
  }
  else {
    if ((*headptr)->next == NULL) {
      (*headptr) = NULL;
      return;
    }
    temp = (*headptr);
    while (temp != NULL) {
      count++;
      temp = temp->next;
    }
    printf ("There are %d elements. Enter position:", count);
    scanf ("%d", &pos);
    if (pos > count+1) {
      printf ("No such position");
      return;
    }
    if (pos == count+1) {
      temp = (*headptr);
      while ((temp->next)->next != NULL)
        temp = temp->next;
      temp->next = NULL;
    }
  else {
    temp = (*headptr);
    while (temp->next != NULL) {
      if (pos == 1) {
        (*headptr) = (*headptr)->next;
        return;
```

```
        if (currpos == pos-1) {
            temp->next = (temp->next)->next;
            return;
        }
        currpos++;
        temp = temp->next;
    }
    printf("No such element found\n");
  }
 }
}
void deletelast (struct node **headptr) {
    struct node *temp;
    temp = (*headptr);
    if ((*headptr) == NULL)
        printf("List is empty\n");
    else if ((*headptr)->next == NULL) {
        (*headptr) = NULL;
    else {
        temp = (*headptr);
        while ((temp->next)->next != NULL)
            temp = temp->next;
        temp->next = NULL;
    }
  }
}
```