ANAGHA ACHARYA
1BM19BT005

classmate
Date
Page

LAB PROGRAM - 6

23.11.20 Singly linked list
a) Create linked list
b) Insertion of node at 1st position, any position end of list
c) Deletion of 1st ele, specified element & last ele.
d) Display contents of linked list

```
#include <stdio.h>
#include < stdlib.h>
void create ();
void display();
void insert_begin();
void insert_end();
void insert_pos();
void delete_begin();
void delete_end();
void delete_pos();

struct node
 {
   int info;
   struct node *next;
 };
struct node *start = NULL;
int main()
{
  int choice;
  while(1) {
     printf ("\n***MENU***\n");
     printf ("~~crea~~\n 1. Create a list");
     printf ("\n 2. Display list");
     printf ("\n 3. Insert node at beginning");
     printf ("\n 4. Insert node at end");
     printf ("\n 5. Insert node at specified position");
printf ("\n 6. Delete node from beginning");
printf ("\n 7. Delete from end");
printf ("\n 8. Delete from specified position");
printf ("\n9. Exit");
printf ("Enter your choice");
scanf ("%d,&choice);
switch ( choice)
{
  case 1:  create();
           break;
  case 2:  display();
           break;
  case 3:  insert_begin();
           break;
  case 4:  insert_end();
           break;
  case 5:  insert_pos();
           break;
  case 6:  ~~insert~~ delete_begin();
           break;
  case 7:  delete_end();
           break;
  case 8:  delete_pos();
           break;
  case 9:  exit(0);
           break;
  default: printf("\nWrong Choice");
 }
}
return 0;
}
```

```
void create()
{
   struct node *temp, *ptr;
   temp = (struct node *) malloc( sizeof(struct node));
   printf ("\n Enter value for node ");
   scanf ("%d", &temp→info);
   temp → next = NULL;
   if (start == NULL)
   {
      start = temp;
   }
   else
   {
      ptr = start;
      while(ptr →next != NULL)
         ptr = ptr→next;
      ptr → next = temp;
   }
}
void display()
{
  struct node *ptr;
  if (start == NULL)
  {
     printf ("\nEmpty list !\n");
     return;
  }
 else
 {
    ptr = start ;
    printf ("\n List elements are:\n");
    while (ptr != NULL) {
       printf ("%d ", ptr→info);
       ptr = ptr→next;
    }
```

```
        }
    }
void insert_begin()
{
    struct node *temp;
    temp = ~~ma~~ (struct node *)malloc(sizeof(struct no
    printf("\n Enter value for node :");
    scanf("%d", &temp->info);
    temp->next = NULL;
    if(start == NULL)
        start = temp;
    else {
        temp -> next = start;
        start = temp;
    }
}
void insert_end(){
    struct node *temp, *ptr;
    temp = (struct node *)malloc(sizeof(struct node));
    printf("\n Enter value for node :");
    scanf("%d", &temp->info);
    temp->next = NULL;
    if(start == NULL)
        start = temp;
    else {
        ptr = start;
        while(ptr->next != NULL)
            ptr = ptr->next;
        ptr->next = temp
    }
}
```

```
void insert_pos() {
    struct node *ptr, *temp;
    int i, pos;
    temp= (struct node *) malloc (sizeof(struct node));
    printf ("\nEnter position for new node:");
    scanf ("%d", &pos);
    printf ("\n Enter value of node:");
    scanf (" %d, &temp→ info);
    temp → next =NULL;
    if (pos= =0)
    {
        temp→ next = start;
        Start =temp;
    }
    else {
     {   for (i=0; ~~i<=~~ ptr= start; i <pos-1; i++)
                ptr = ptr → next;
            temp→next = ptr → next;
            ptr → next = temp;
        }
    }
void delete_begin() {
    struct node *ptr;
    if (start= =NULL)
    {
        printf ("Empty list!\n");
        return;
    }
    else
    {
        ptr= start;
        start= start→ next;
        printf ("\n Deleted element is :%d", ptr→info);
    } free(ptr);
}
```

```
void delete end() {
{ struct node *ptr, ~~*temp~~;
   if (start==NULL)
   {
       printf("\n Empty list");
       exit(0);
   }
  else if (start->next ==NULL)
  {
     ptr = start;
     start = NULL;
     printf("\n Deleted element is :%d", ptr->info);
     free(ptr);
  }
 else {
    ptr = start;
    while(ptr->next ! =NULL)
     {
        temp= ptr;
        ptr = ptr->next;
     }
    temp->next = NULL;
    printf("Deleted element is :%d", ptr->info);
    free(ptr);
 }
}
void delete pos() {
   int i, pos;
   struct node *temp, *ptr;
   if(start ==NULL)
  { printf("\n Empty list !");
     exit(0)
   }
   else {
```

```
printf ("\nenter the position of node \n");
scanf ("%d", &pos);
if (pos == 0)
{
    ptr = start ;
    start = start -> next ;
    printf ("\n Deleted element is %d : ptr -> info);
    free(ptr);
}
else {
    ptr = start ;
    ~~star~~ for(i=0; i<pos ; i++)
    {
        temp = ptr ;
        ptr = ptr -> next ;
          if (ptr == NULL)
          {
             printf ("\n Position not found \n");
             return ;
          }
    }
    temp -> next = ptr -> next ;
    printf ("Deleated element is %d ", ptr -> info");
    free(ptr);
  }
 }
}
```