Divyashree.k
1BM19CS054

Lab 7

```
#include <stdio.h>
#include <stdlib.h
#include <string.h>
struct node
{
  int sum;
  struct node* next;
};
struct node* head1=NULL;
struct node* head2=NULL;
int c=0;
void Insert()
{
    struct node* newnode;
     struct node* temp;
    int s;
    printf("Enter integer:");
     scanf("%d",&s);
     newnode=(struct node*)malloc(sizeof
                            ((struct node));
   newnode->sum=s;
    if (head==NULL)
    {
     newnode->next=NULL;
     head=newnode;
     printf("first node of linked list created\n");
       c++;
    }
else
  {
      temp=head;
      while(temp->next!=NULL)
     {
          temp=temp->next;
     }
        temp->next=newnode.
```

Divyashree.k

```
newnode -> next =NULL;
 c++;
printf("Node created \n");
}
}

void Insert2()
 {
    struct node* newnode;
    struct node* temp;
    int s, y;
    printf ("Enter-elements to create list 2 \n");
    do
    {
    printf(" Enter integer :\n");
    scanf("%d", &s);
    newnode = (struct node*)malloc(sizeof
                                 (struct node));

    newnode -> sum = s;
    if ( head 2 = = NULL)
    {
     newnode -> next =NULL;
     head 2 = newnode;
     printf (" first node of linked list
                created \n");
     C~~a~~++;
    }
    else
    {
        temp =head2;
        while (temp -> next != NULL)
        {
           temp = temp -> next;
        }
        temp -> next ~~=NULL~~=newnode;
        newnode -> next =NULL;
        c++;
```

L

```
printf("Node created \n");
}
 printf(" do you want to continue adding:0 or
                                   1\n");
 scanf("%d", &y);
} while(y!=0);
}
 void bubbleSort()
 {
   int swapped,i;
   struct node *ptr1;
   struct node *lptr = NULL;
   if (head==NULL)
     return;
   do
     { swapped=0;
     prt 1= head;
      while (ptr 1-> next != lptr)
     {
      if (ptr->sum>ptr1->next->sum)
      {
        int temp = ptr 1->sum;
        ptr 1-sum=ptr->next->sum;
        ptr 1-snext->sum=temp;
         swapped=1;
      }
        prt=ptr1->next;
     }
     lptr=ptr 1;
   }
   while (swapped);
 }
 void reverse()
 {
   struct node* prev=NULL;
struct node* current = head;
struct node* next = NULL;
while (current != NULL){
next = current ->next;
current-> next = prev;
prev = current;
current = next; }
 head = prev;
}

void concat ()
{
  struct node *ptr;
  if ( head == NULL)
  {
     head = head2;
  }
  if(head 2 == NULL)
  {
     head2 = head;
  }
  ptr = head;
 while (ptr->next != NULL)
        ptr = ptr->next;
   ptr ->next = head 2;
}
void Display1()
{
  struct node *ptr;
  ptr = head;
   int i = 1;
   if (ptr == NULL)
   { printf ("Linked list is empty!\n");
 } else
 {
   while (ptr != NULL)
   {
```

Duryodhan.K

```
      printf("%d", ptr->sum);
      i++;
      ptr = ptr->next
    }
  }
}

void display2()
{
   struct node *ptr;
   ptr = head2
   int i=1;
   if (ptr==NULL)
   {
     printf("Linked list is empty!\n");
   }
   else{
   while (ptr!=NULL)
   {
     printf("%d", ptr->sum);
     printf("\n");
    i++;
     ptr=ptr->next
   }
  }
}

int main()
{
int choice,pos;
do {
printf("\n1.Insert node \n2.sort
          nod \n3.reverse node\n4.
          concat 2 list \n5.exit\n");
 printf("\n Enter your choice:");
 scanf("%d", &choice);
  switch (choice)
   {
```

```
case1 :
Insert ( )
break ;
case2 :
bubble sort ();
display ();
break;
 case 3:
 reverse()
  display ( );
 break ;
 case 4 :
 Insert  2( );
  concate ( );
   display ( );
  break
  case 5 :
  break;
  default :
   printf ("Wrong choice\n");
   break;
  }
 } while (choice!=5);
 return 0;
}
```