IBM19CS006

27/11/20 Lab 7 - Sort, Reverse, Concat List

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

```
newnode->next=NULL
    C++;
  }
}

void Insert2() {
struct node *temp, *newnode;
  int n, y;
  printf("Enter elements to create list 2\n");
  do {
    printf("Enter integer : \n");
    scanf("%d",&n);
newnode=(struct node*)malloc(sizeof(struct node));
newnode->data=n;
if(head2==NULL)
{   newnode->next=NULL
    head2=newnode;
    C++;
  }
else {
  temp=head2;
while(temp->next!=NULL)
{   temp=temp->next;
  }
  temp->next=newnode;
  newnode->next=NULL;
    C++;
  }
printf("Press any number to continue creating list
```

```
& 0 to terminate \n");
scanf("%d", &y);
}while(y!=0);
}

void Sort()
{ int swap,i;
struct node *ptr1, *lptr=NULL;
if(head == NULL)
    return;
do{
  swap=0;
  ptr1=head;
while(ptr1->next!=lptr){
  if(ptr1->data > ptr1->next->data)
{
  int temp = ptr1->data;
  ptr1->data = ptr1->next->data;
  ptr->next->data=temp;
  swap=1;
}
ptr1 = ptr1=> next;
}
  lptr = ptr1;
}
swap while(swap);
}
```

```
void Reverse() {
struct node * prev = NULL, *current = head; *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 (head2 == NULL)
{  head2 = head;
   }
  ptr = head;
  while ( ptr -> next != NULL)
       ptr = ptr -> next;
  ptr -> next = head2;
}
void display1()
{  struct node * ptr;
   ptr = head;
    int i = 1;
```

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

```
int main()
{
  int choice, pos;
  do{
  printf("\n1.Insert \n2. Sort \n3. Reverse \n4.
  Concatenate 2 lists \n5. Exit \n");
  scanf("%d", &choice);
  switch(choice)
{
  case 1: Insert();
          break;
  case 2: Sort();
          display1(); break;
  case 3: Reverse();
          display1();
          break;
  case 4: Insert2();
          concat();
          display1(); break;
  case 5: exit(0);
          break;
  }
}while(choice!=5);
 return 0;
}
```