ADITHI GURIMAJI
1BM19CS005

Lab-7

```
#include<stdio.h>
#include<stdlib.h>
void create();
void display();
void sort();
void reverse();
void concatenate();
struct node
{ int data;
  struct node *next;
};
struct node *head=NULL;
int main(int argc, char **argv)
{ int choice, ch;
  do{
  printf("1.Create\n2.Display\n3. Sort in ascending order
\n4. Reverse \n5. Concatenate 2 linked lists\n");
  printf("\n Enter your choice:");
  scanf("%d", &choice);
  switch(choice)
{ case 1: create(); break;
  case 2: display(); break;
  case 3: sort(); break;
  case 4: reverse(); break;
  case 5: concatenate(); break;
  default: printf("Wrong choice");
}
printf("\n Press 1 if you want to continue else any
other number\n");
scanf("%d", &ch);
} while(ch==1); return 0; }
```

```
void create()
{ struct node *newnode, *temp;
  int item;
  newnode = (struct node *) malloc (sizeof(struct node));
  printf("Enter the data:");
  scanf("%d", &item);
  newnode->data=item;
  if (head==NULL)
  { newnode->next=NULL;
    head=newnode;
    printf("Node is created\n");
  } else
  { temp=head;
    while(temp->next!=NULL)
  { temp=temp->next; }
    temp->next=newnode;
    newnode->next=NULL;
    printf("Node is created\n");
}}
void display()
{ struct node *ptr;
  ptr=head;
  if (ptr==NULL)
  printf("Nothing to print\n");
  else
  { while(ptr!=NULL)
    { printf("%d", ptr->data);
      ptr=ptr->next;
  }}}
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 sort()
{ int swapped, temp;
  struct node *ptr1;
  struct node *lptr = NULL;
  if (head == NULL)
  { printf ("List is empty\n");
    return;
  }
  if (head->next == NULL)
  { printf ("Sorted list:\n");
    printf("%d\n", head -> data); return;
  }
  do {
  swapped = 0; ptr1 = head;
  while (ptr1->next != lptr)
  {
    if (ptr1 -> data > ptr1->next->data) {
      temp = ptr1->data;
      ptr1 -> data = ptr1->next-> data;
      ptr1 -> next -> data = temp;
      swapped = 1; }
      ptr1 = ptr1 -> next; }
      lptr = ptr1; }
      while (swapped);
}
```

```
void concatenate()
{ struct node *head1=NULL;
  struct node *head2=NULL;
  struct node *newnode1, *temp1;
  struct node *newnode2, *temp2;
  int item, f=1;
  printf(" ---First list---\n");
  do{
  printf("Enter the data:");
  scanf("%d", &item);
  newnode1=(struct node*)malloc(sizeof(struct node));
  newnode1->data=item;
  if(head1==NULL)
    { newnode1->next=NULL;
      head1=newnode1;
      printf("Node is created\n");
    } else {
    temp1=head1;
        while(temp1->next!=NULL)
        { temp1=temp1->next; }
        temp1->next=newnode1;
        newnode1->next=NULL;
        printf("Node is created\n");
  } printf("Press 1 if you want to continue, else any
  other number\n");
  scanf("%d", &f);
  } while(f==1);
  printf("--- Second list ---\n");
  do{
  printf("Enter the data:");
  scanf("%d", &item);
  newnode2=(struct node*) malloc(sizeof(struct node));
```

```
newnode2 -> data = item;
if(head2 == NULL)
{   newnode2 -> next = NULL;
    head2 = newnode2; printf("Node is created\n");
}

 else {
temp2 = head2;
   while(temp2 -> next != NULL)
   { temp2 = temp2 -> next; }
    temp -> next = newnode2;
   newnode2 -> next = NULL;
   printf("Node is created\n"); }
printf("Press 1 to continue else any other number\n");
 scanf("%d", &f); } while (f == 1);
temp1 = head1;
while(temp1 -> next != NULL)
{  temp1 = temp1 -> next; }
 temp1 -> next = head2;
printf("Concatenated list: \n");
 struct node *ptr1;
ptr1 = head1;
if (ptr1 == NULL)
{ printf("Nothing to print\n");
}
else
{  while(ptr1 != NULL)
    { printf("%d\t", ptr1 -> data);
      ptr1 = ptr1 -> next;
    }
} }
```