1.WAP to Implement Single Link List with following operations: Sort the linked list, Reverse the linkedlist, Concatenation of two linked lists.

Shashank Patel C J

1BM22CS255

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
   int data;
   struct node *next;
};
void insertatbegin(struct node** head,int value)
{
   struct node* new_node=(struct node*)malloc(sizeof(struct node));
   new_node->data=value;
   new_node->next=*head;
   *head=new_node;
}
void concat(struct node *head1,struct node *head2)
{
   if (head1->next == NULL)
      head1->next = head2;
   else
```

```
        concat(head1->next,head2);
}
void sortlist(struct node** head1)
{
   struct node *temp,*i;
   for(temp=*head1;temp!=NULL;temp=temp->next)
   {
      for(i=temp->next;i!=NULL;i=i->next)
      {
         if(i->data < temp->data)
         {
            int tem=i->data;
            i->data=temp->data;
            temp->data=tem;
         }
      }
   }
}
void reverse(struct node** head1)
{
   struct node *prev=NULL;
   struct node *current=*head1;
   struct node* next=NULL;
   while(current!=NULL)
   {
```

```
        next=current->next;
        current->next=prev;
        prev=current;
        current=next;
    }
    *head1=prev;
}
void printlist(struct node* node)
{
    struct node* temp=node;
    while(temp!=NULL)
    {
        printf("%d-->",temp->data);
        temp=temp->next;
    }
    printf("NULL\n");
}
int main()
{
    struct node *head1=NULL;

    insertatbegin(&head1,10);
    insertatbegin(&head1,15);
    insertatbegin(&head1,40);
    insertatbegin(&head1,50);
```

```
    printf("List 1:");
    printlist(head1);

    struct node *head2=NULL;
    insertatbegin(&head2,65);
    insertatbegin(&head2,75);
    insertatbegin(&head2,60);

    printf("List 2:");
    printlist(head2);

    concat(head1,head2);
    printf("List after concatenation:");
    printlist(head1);

    sortlist(&head1);
    printf("List after sorting:");
    printlist(head1);

    reverse(&head1);
    printf("Reversed Linked list");
    printlist(head1);
}
```

Output:

```
List 1:50-->40-->15-->10-->NULL
List 2:60-->75-->65-->NULL
List after concatenation:50-->40-->15-->10-->60-->75-->65-->NULL
List after sorting:10-->15-->40-->50-->60-->65-->75-->NULL
Reversed Linked list75-->65-->60-->50-->40-->15-->10-->NULL

Process returned 0 (0x0)   execution time : 0.109 s
Press any key to continue.
```