WAP TO SORT,REVERSE AND CONCAT TWO LISTS

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

struct node
{
   int data;
   struct node *next;
};
void insertatbeg(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;

   insertatbeg(&head1,10);
   insertatbeg(&head1,90);
   insertatbeg(&head1,20);
   insertatbeg(&head1,80);
   printf("List 1:");
   printlist(head1);

   struct node *head2=NULL;
   insertatbeg(&head2,30);
   insertatbeg(&head2,70);
   insertatbeg(&head2,40);
```

```
    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: