WAP to implement Singly linked list with following operations
(a) Sort the linked list
(b) Reverse the linked list
(c) Concatenation of 2 linked list

```
void sort (struct node **head)
{ struct node  * temp1 = *head;
   struct node  * temp2 = *head;
     if (temp1 == NULL)
        printf (list is empty)
     else if (temp1→next == NULL)
        printf (" only element in the list /-/n"
                        temp1→data);
   else
     {  int a;
        for (temp1 = *head; temp1!= NULL;
               temp = temp1→ next)
        {  for (temp2 = temp1→next; temp2! =
                NULL; temp2 = temp2→next)
           { if ((temp1→data) > temp2→data))
             { int a = temp1→data;
               temp1→data = temp2 → data;
               temp2→data = a;
             }
        }
     }
   }
   display sorted list (head);
}
```

```c
display(struct node **head)
{
    struct node *temp = *head;
    printf("Sorted elements are:");
    while(temp != NULL)
    {
        printf(" %d", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

void reverse(struct node **head)
{
    struct node *prev = NULL
    struct node *next = NULL;
    while(head != NULL)
    {
        next = *head->next;
        *head->next = prev;
        prev = *head;
        head = next;
    }
    *head = prev;
}

display(**
```

```c
struct node * concat (struct node **head
                      struct node **head2)
{
    struct node *ptr;
    if (*head == NULL)
        {
            *head = *head2;
        }
    if (head2 == NULL)
        return *head;

    ptr = *head;
    while ( ptr->next != NULL)
        ptr = ptr->next;
    ptr->next = *head2
        return *head;
}
```