Sort, Reverse, Concatenate

```
void sort (struct node *hptr)
{
   if (hptr == NULL)
      printf ("Empty list \n");
   else
   {
     int swap;
     struct node *first = NULL;
     struct node *last = NULL;
     do
     {
         swap = 0;
         first = hptr;
      while (first -> next != last)
      {
        if (first -> data > first -> next -> data)
        {
            int temp = first -> data;
            first -> data = first -> next -> data;
            first -> next -> data = temp;
            swap = 1;
        }
        first = first -> next
      }
        last = first;
    } while (swap);
  }
}
```

```
void reverse (struct node **hptr)
{
  if (*hptr == NULL)
   {
     printf("Empty list \n");
   }
   else
  {
   struct node *prev, *curr, *head = *hptr
     prev = head;
     curr = head-> next;
      head = head-> next;
      prev ->next = NULL;
     while (head != NULL)
      {
        head = head -> next;
        curr -> next = prev;
          prev = curr;
          curr = head;
      }
       *hptr = prev;
    }
}
```

```
void concatenate (struct node *hptr1, struct
                                node *hptr2)
{
    if (hptr1==NULL && hptr2==NULL)
         printf ("Both are empty list \n");
      else if (hptr1==NULL || hptr2==NULL)
         printf ("One of them is empty\n");
      else
     {
       struct node* temp1= hptr1;
       struct node* temp2 = hptr2;
       while (temp1-> next!= NULL)
             temp1 = temp1 -> next;
             temp1 -> next = temp2
      }
 }
```