Date
Page

```
#include <stdlib.h>
#include <string.h>
struct node
{
        int sem;
        struct node *next;
};
struct node *head1 = NULL;
struct node *head2 = NULL;
int c=0;

void insert()
{
        struct node* newnode;
        struct node* temp;
        int s;
        printf("Enter integer:");
        scanf("%d", &s);
        newnode=(struct node*)malloc(sizeof(struct node));
        newnode->sem=s;
        if(head==NULL)
        {
                newnode->next=NULL;
                head=newnode;
                printf("first node of linked list created\n");
                c++;
        }
        else
        {
                temp=head;
                while(temp->next!=NULL)
                {
                        temp=temp->next;
        }
        temp->next = newnode;
        newnode->next = NULL;
        c++;
        printf("Node created\n");
      }
   }

void Insert2()
{
     struct node* newnode;
     struct node* temp;
     int i,y;
     printf("Enter ellement to create list 2\n");
     do
     {
          printf("Enter integer:\n");
          scanf("%d", &s);
          newnode = (struct node*) malloc(sizeof(struct node*));
          newnode->sem = s;
          if (head2 == NULL)
          {
               newnode->next = NULL;
               head2 = newnode;
               printf("first node of linked list created\n");
               c++;
          }
          else
          {
               temp = head2;
               while (temp->next != NULL)
               {
                    temp = temp->next;
               }
               temp->next = newnode;
```

```
newnode -> next = NULL;
c++;
printf("Node Created\n");
}
printf("do u want to continue adding: 0 or 1 \n");
scanf("%d", &y);
}
while(y!=0);
}
void bubblesort()
{
    int swapped, i;
    struct node* ptr1;
    struct node* lptr= NULL;

    if(head==NULL)
      return;

    do
    {
        swapped 0;
        ptr1 = head;
        while( ptr1 -> next != lptr)
        {
            if (ptr1 -> sem > ptr1 -> next -> sem)
            {
                int temp = ptr1 -> sem;
                ptr1 -> sem = ptr1 -> next -> sem;
                ptr1 -> next -> sem = temp;
                swapped = 1;
            }
        ptr1 = ptr1 -> next;
        }
```

```
}ptr = ptr1;
}
while(swapped);
}
void reverse()
{
        struct node* prev = NULL;
        struct node* ~~prev~~ current = head;
        struct node* next = NULL;
        while( current != NULL),
        {
            next = current -> next;
            current-> next = prev;
             prev = current;
             current= next;
        }
            head = prev;
 }
 void concat()
  {
        struct node* ptr;
        if (head == NULL)
         {
              head = head2;
         }
         if( head2 == NULL)
           {
              head2 = head;
             }
           ptr = head;
           while (ptr -> next != NULL)
                   ptr = ptr -> next;
                ptr -> next = head2;
    }
```

```
void display1()
{
    struct node* ptr;
    ptr = head;
     int i=1;
    if (ptr==NULL)
    {
       printf(" Linked list is empty \n");
    }
    else
    {
        while (ptr!=NULL)
        {
            printf(" %d", ptr -> rem);
            i++;
            ptr = ptr -> next;
        }
    }
}
int main()
{
    int choice, pos;
    do
    {
        printf(" \n1. Insert node \n2. sort node \n3
         . reverse node ptr4. concat 2list \ns
          exit \n");
        printf("\n Enter your choice");
        scanf(" %d, &choice);
        switch(choice)
        {
            case 1:
            insert();
            break;
case 2:
 printf("before:\n");
  display 1();
  bubblesort ();
   printf(" after:\n");
   display1();
   break;

 case3:
   printf ("before:\n");
    display1();
    reverse ();
    printf ("after:\n");
     display1();
      break;

   case 4:
   insert 2();
   concat ();
    display3();
     break;

      case 5:
      break;

       default:
       printf("wrongchoice");
        break;
    }
    while(choice!=5);
    }
    return0;
}
```