// Program to sort & reverse a given linked list.

```
void create();
void display();
void sort();
void reverse();

struct node
{ int data;
  struct node *next;
};
struct node *head=NULL;

int main()
{
   int choice, ele;
   do {
    printf("1.Create 2. Display 3. Sort 4- Reverse 5. Exit \n");
    printf("enter your choice");
    switch(choice)
    {
          case 1 : create(); break;
          case 2 : display(); break;
          case 3 : sort(); break;
          case 4 : reverse(); break;
          default : exit(0);
    }
   } while (choice==1 || choice==2 || choice==3 || choice==4);
   return 0;
}

void create()
{
   struct node *newnode, *temp;
    int item;
    newnode = (struct node*) malloc (sizeof (struct node));
     printf("enter data");
     scanf("%d", &item);
     newnode->data = item;
     if (head==NULL)
       {
          newnode->next = NULL;
           head = newnode;
           printf("Node created\n");
       }
```

```
else {
        temp=head;
        while(temp->next!=NULL)
            temp=temp->next;
        temp->next=newnode;
        newnode->next=NULL;
    }   }

void display()
{
    struct node *ptr=NULL;
    ptr=head;
    if(ptr==NULL)
    {
        printf("empty");
    }
    else {
        while(ptr!=NULL)
        { printf("%d", ptr->data);
          ptr=ptr->data.next;
        }   }
}

void sort()
{
    struct node *temp, *ptr, *ptr2;
    int a;
    temp=head;
    ptr=head;
    for(temp=head; temp!=NULL; temp=temp->next)
    {
        for(ptr=temp; ptr!=NULL; ptr=ptr->next)
        {
            if(ptr->data < temp->data)
            {
                a=temp->data;
                temp->data=ptr->data;
                ptr->data=a;
            }   }   }   }

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

// Concatenation of two list by local pointer.

```
#include<stdio.h>
struct node { int data;
              struct node *next;
              };
void create(struct node**);
void display(struct node*);
void concat(struct node*, struct node**);
int main()
{
    struct node *head1=NULL, *head2=NULL;
    create(&head1);
    create(&head2);
    concat(head1, head2);
    display(head1);
}
void create(struct node **hptr)
{
    struct node *newnode, *temp;
    int item, choice=1;
    do {
        newnode=(struct node*) malloc(sizeof(struct node));
        printf("enter data");
        scanf("%d", &item);
        newnode->data=item;
        newnode->next=NULL;
        printf("Do you want to add element to list (if yes enter 1)");
        scanf("%d", &choice);
        if (*hptr==NULL)
            *hptr=newnode;
        else {
            temp=*hptr;
            while (temp->next!=NULL)
                temp=temp->next;
            temp->next=newnode;
            newnode->next=NULL;
        }
    } while (choice==1);
}
```

```
void concat (struct node *temp1, struct node *temp2)
{
    while (temp1->next != NULL)
        temp1 = temp1->next;
    temp1->next = temp2;
}

void display(struct node *ptr)
{
    if (ptr == NULL)
        printf("nothing to print");
    else {
        while (ptr != NULL)
        {
            printf("%d", ptr->data);
            ptr = ptr->next;
        }
    }
    printf("\n");
}
```