```c
/*Assignment 6:-
Write a c program to perforn sieve sort on a given set of inputs*/

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

/*Node structure*/
struct node
{
    int data;
    struct node *next,*spar,*prev;
};

/*Creating a node*/
struct node *getnode(int x)
{
    struct node *new1;
    new1=(struct node *)malloc(sizeof(struct node));
    new1->data=x;
    new1->next=NULL;
    new1->spar=NULL;
    new1->prev=NULL;
    return(new1);
}

/*Head pointer*/
struct node *head;

/*Main Function*/
void main()
{
    void insert();
    void sieve_sort();
    void display();
    insert();
    printf("\nList Before Sorting\n");
    display();
    sieve_sort();
    printf("\nList After Sorting\n");
    display();
}

/*Function to insert the elements*/
void insert()
{
    int x;
    struct node *new1,*ptr;
    head=NULL;
    do
    {
        printf("Enter element :- ");
        scanf("%d",&x);
        new1=getnode(x);
        if(head == NULL)
            head=new1;
        else
        {
            ptr=head;
            while(ptr->next != NULL)
                ptr=ptr->next;
            ptr->next=new1;
        }
        printf("Do you want to enter any more element?(0=YES,1=NO) :- ");
        scanf("%d",&x);
    }while(x==1);
}

/*Function to perform the sorting*/
void sieve_sort()
{
    struct node *head2,*head3,*ptr1,*ptr2,*a[100],*new1,*ptr;
    int f1=1,f2,i,j;
    /*1st pass:-to traverse the initial list of elements from left to right and partition them
 accordingly*/
    head2=head;
    head=head->next;
    head2->next=NULL;
    while(head != NULL)
    {
        if(f1 == 1)
        {
            ptr2=head2;
            ptr1=head;
        }
        if(ptr2->data > ptr1->data)
        {
            while(ptr2->spar != NULL)
                ptr2=ptr2->spar;
            ptr2->spar=ptr1;
            head=head->next;
            ptr1->next=NULL;
            ptr1->prev=ptr2;
            f1=1;
        }
        else
        {
            if(ptr2->next == NULL)
            {
                ptr2->next=ptr1;
                head=head->next;
                ptr1->next=NULL;
                f1=1;
            }
            else
            {
                ptr2=ptr2->next;
                f1=0;
            }
        }
    }
    head3=head2;
    /*next passes:-to traverse the list of elements formed after the first pass from left to r
ight and partition them accordinly*/
    for(i=0;i<10;i++)
        a[i]=NULL;
    do
    {
        i=0;
        while(head3 != NULL)
        {
            head=head3;
            head3=head3->next;
            while(head->spar != NULL)
            {
                ptr1=head;
                head=head->spar;
                ptr1->spar=NULL;
            }
            new1=getnode(head->data);
            ptr=head;
            head2=new1;
            head=head->prev;
            free(ptr);
            f1=1;
            while(head != NULL)
            {
                if(f1 == 1)
                {
                    ptr2=head2;
                    ptr1=head;
                }
                if(ptr2->data > ptr1->data)
                {
                    while(ptr2->spar != NULL)
                        ptr2=ptr2->spar;
                    new1=getnode(ptr1->data);
                    ptr=ptr1;
                    ptr2->spar=new1;
                    new1->prev=ptr2;
                    head=head->prev;
                    free(ptr);
                    f1=1;
                }
                else
                {
                    if(ptr2->next == NULL)
                    {
                        new1=getnode(ptr1->data);
                        ptr=ptr1;
                        ptr2->next=new1;
                        head=head->prev;
                        free(ptr);
                        f1=1;
                    }
                    else
                    {
                        ptr2=ptr2->next;
                        f1=0;
                    }
                }
            }
            a[i++]=head2;
        }
        /*updating the list after the partitions have bben made so that all the partitions are
 together*/
        for(j=0;j<i-1;j++)
        {
            head2=a[j];
            while(head2->next!=NULL)
                head2=head2->next;
            head3=a[j+1];
            head2->next=head3;
        }
        head3=a[0];
        ptr1=head3;
        f2=0;
        /*checking if all the elements have been sorted in the required order*/
        while(ptr1 != NULL)
        {
            if(ptr1->spar != NULL)
            {
                f2=1;
                break;
            }
            ptr1=ptr1->next;
        }
    }while(f2 == 1);
    head=head3;

}

/*Function to display the elements*/
void display()
{
    struct node *ptr;
    ptr=head;
    printf("\n");
    while(ptr != NULL)
    {
        printf("%d\t",ptr->data);
        ptr=ptr->next;
    }
    printf("\n");
}
```