



```
38  
39 }  
40 }  
41 void Insert2()  
42 {  
43     struct node *newnode;  
44     struct node *temp;  
45     int s,y;  
46  
47     do  
48     {  
49         printf("Enter integer : \n");  
50         scanf("%d",&s);  
51         newnode=(struct node*)malloc(sizeof(struct node));  
52         newnode->sem =s;  
53         if (head2==NULL)  
54         {  
55             newnode->next=NULL;  
56             head2=newnode;  
57  
58             c++;  
59         }  
60         else  
61         {  
62             temp=head2;  
63             while(temp->next!=NULL)  
64             {  
65                 temp=temp->next;  
66             }  
67             temp->next=newnode;  
68             newnode->next=NULL;  
69             c++;  
70  
71         }  
72         printf("do u want to continue adding:0 or 1\n");  
73         scanf("%d",&y);
```



```

74     }while(y!=0);
75 }
76
77
78 void bubbleSort()
79 {
80     int swapped, i;
81     struct node *ptr1;
82     struct node *lptr = NULL;
83
84
85     if (head == NULL)
86         return;
87
88     do
89     {
90         swapped = 0;
91         ptr1 = head;
92
93         while (ptr1->next != lptr)
94         {
95             if (ptr1->sem > ptr1->next->sem)
96             {
97                 int temp = ptr1->sem;
98                 ptr1->sem = ptr1->next->sem;
99                 ptr1->next->sem = temp;
100                 swapped = 1;
101             }
102             ptr1 = ptr1->next;
103         }
104         lptr = ptr1;
105     }
106     while (swapped);
107 }
108
109 void reverse()
110 {

```



```
103     }
104     lptr = ptr1;
105 }
106 while (swapped);
107 }
108
109 void reverse()
110 {
111     struct node* prev = NULL;
112     struct node* current = head;
113     struct node* next = NULL;
114     while (current != NULL) {
115         next = current->next;
116         current->next = prev;
117         prev = current;
118         current = next;
119     }
120     head = prev;
121 }
122
123 void concat()
124 {
125     struct node *ptr;
126     if(head==NULL)
127     {
128         head=head2;
129     }
130     if(head2==NULL)
131     {
132         head2=head;
133     }
134     ptr=head;
135     while(ptr->next!=NULL)
136         ptr=ptr->next;
137     ptr->next=head2;
138 }
```



```

135     while(ptr->next!=NULL)
136         ptr=ptr->next;
137     ptr->next=head2;
138 }
139 void display1()
140 {
141     struct node *ptr;
142     ptr=head;
143     int i=1;
144
145     if(ptr==NULL)
146     {
147         printf("Linked list is empty!\n");
148     }
149     else
150     {
151         while(ptr!= NULL)
152         {
153             printf(" %d",ptr->sem);
154             i++;
155             ptr=ptr->next;
156         }
157     }
158 }
159
160 }
161 void display2()
162 {
163     struct node *ptr;
164     ptr=head2;
165     int i=1;
166
167     if(ptr==NULL)
168     {
169         printf("Linked list is empty!\n");
170     }
171     else

```



```

160 }
161 void display2()
162 {
163     struct node *ptr;
164     ptr=head2;
165     int i=1;
166
167     if(ptr==NULL)
168     {
169         printf("Linked list is empty!\n");
170     }
171     else
172     {
173         while(ptr!= NULL)
174         {
175
176
177             printf(" %d",ptr->sem);
178             printf("\n");
179             i++;
180             ptr=ptr->next;
181         }
182     }
183 }
184
185 }
186
187 int main()
188 {
189     int choice,pos;
190     printf("\n1. Insert \n2. sort \n3. reverse \n4.concat 2 lists \n5.exit\n");
191     do
192     {
193
194
195         printf("\nEnter your choice : ");

```



```

184 }
185 }
186
187 int main()
188 {
189     int choice,pos;
190     printf("\n1. Insert \n2. sort \n3. reverse \n4.concat 2 lists \n5.exit\n");
191     do
192     {
193
194
195         printf("\nEnter your choice : ");
196         scanf("%d",&choice);
197         switch(choice)
198         {
199             case 1:
200                 Insert();
201                 break;
202
203             case 2:
204                 bubbleSort();
205                 display1();
206                 break;
207
208             case 3:
209                 reverse();
210                 display1();
211                 break;
212
213             case 4:
214                 Insert2();
215                 concat();
216                 display1();
217                 break;
218
219             case 5:
220                 break;

```



```
193
194
195     printf("\nEnter your choice : ");
196     scanf("%d",&choice);
197     switch(choice)
198     {
199         case 1:
200             Insert();
201             break;
202
203         case 2:
204             bubbleSort();
205             display1();
206             break;
207
208         case 3:
209             reverse();
210             display1();
211             break;
212
213         case 4:
214             Insert2();
215             concat();
216             display1();
217             break;
218
219         case 5:
220             break;
221
222         default:
223             printf("Wrong choice!\n");
224             break;
225     }
226 }while(choice!=5);
227 return 0;
228 }
```





```
1. Insert
2. sort
3. reverse
4.concat 2 lists
5.exit

Enter your choice : 1
Enter integer : 12

Enter your choice : 1
Enter integer : 6

Enter your choice : 1
Enter integer : 45

Enter your choice : 2
6 12 45
Enter your choice : 3
45 12 6
Enter your choice : 4
Enter integer :
34
do u want to continue adding:0 or 1
1
Enter integer :
67
do u want to continue adding:0 or 1
0
45 12 6 34 67
Enter your choice : 2
6 12 34 45 67
Enter your choice : █
```

