

```
age1  Generics.java x del.c a1.c x dept1.css x
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  struct node
5  {
6      int sem;
7      char name[50];
8      char usn[50];
9      struct node *next;
10 };
11 struct node *head= NULL;
12 int c=0;
13 void Insert()
14 {
15     struct node *newnode;
16     struct node *temp;
17     int s;
18     char n[30],u[30];
19     printf("Enter your name  : ");
20     scanf("%s",n);
21     printf("Enter your semester : ");
22     scanf("%d",&s);
23     printf("Enter your usn  : ");
24     scanf("%s",u);
25     newnode=(struct node*)malloc(sizeof(struct node));
26     newnode->sem =s;
27     strcpy(newnode->name,n);
28     strcpy(newnode->usn,u);
29     if (head==NULL)
30     {
31         newnode->next=NULL;
32         head=newnode;
33         printf("first node of linked list created\n");
34         c++;
35     }
36     else
37     {
38         temp=head;
39         while(temp->next!=NULL)
40         {
41             temp=temp->next;
42         }
43         temp->next=newnode;
```

```
age1 Generics.java x del.c a1.c x dept1.css x Student.java — Desktop x Internals.java
36     else
37     {
38         temp=head;
39         while(temp->next!=NULL)
40         {
41             temp=temp->next;
42         }
43         temp->next=newnode;
44         newnode->next=NULL;
45         c++;
46         printf("Node created\n");
47     }
48 }
49 void deletebeg()
50 {
51     struct node *ptr;
52     if(head == NULL)
53     {
54         printf("\nList is empty");
55     }
56     else
57     {
58         ptr = head;
59         head = ptr->next;
60         free(ptr);
61         printf("\n Node deleted from the beginning ...");
62     }
63 }
64 void deletemid()
65 {
66     char key[20];
67     printf("enter the usn of student to be deleted\n");
68     scanf("%s",key);
69     struct node *temp = head, *prev;
70     if (temp != NULL && strcmp(temp->usn,key)==0)
71     {
72         head = temp->next;
73         free(temp);
74         return;
75     }
76     while (temp != NULL && strcmp(temp->usn,key)!=0)
77     {
```

```
age1 Generics.java x del.c a1.c x dept1.css x Student.java — Desktop x Internals.java
76     while (temp != NULL && strcmp(temp->usn,key)!=0)
77     {
78         prev = temp;
79         temp = temp->next;
80     }
81
82     if (temp == NULL)
83     {
84         printf("student not in the list\n");
85         return;
86     }
87     prev->next = temp->next;
88
89     free(temp);
90 }
91 void deleteend()
92 {
93
94     struct node *toDelLast, *preNode;
95     if(head == NULL)
96     {
97         printf(" There is no element in the list.");
98     }
99     else
100     {
101         toDelLast = head;
102         preNode = head;
103         while(toDelLast->next != NULL)
104         {
105             preNode = toDelLast;
106             toDelLast = toDelLast->next;
107         }
108         if(toDelLast == head)
109         {
110             head = NULL;
111         }
112         else
113         {
114             preNode->next = NULL;
115         }
116         free(toDelLast);
117     }
```

```
age1  Generics.java x del.c a1.c x dept1.css x Studer
119
120 void display()
121 {
122     struct node *ptr;
123     ptr=head;
124     int i=1;
125
126     if(ptr==NULL)
127     {
128         printf("Linked list is empty!\n");
129     }
130     else
131     {
132         while(ptr!= NULL)
133         {
134             printf("----NODE %d----\n",i);
135             printf("Name: %s\n",ptr->name);
136             printf("USN: %s\n",ptr->usn);
137             printf("Sem: %d\n",ptr->sem);
138             printf("\n");
139             i++;
140             ptr=ptr->next;
141         }
142     }
143 }
144
145 }
146
```

```
age1  Generics.java x del.c a1.c x dept1.css x Student.java — Desktop x Internals.java x Externals.java x Main1.java x .gitignore x ? x
140
147 int main()
148 {
149     int choice,pos;
150     do
151     {
152
153
154         printf("\n1. Insert node \n2. delete node in the beg of the list\n3. delete at the end of list\n4.delete a given node \n5. display list\n6.exit\n");
155         printf("\nEnter your choice : ");
156         scanf("%d",&choice);
157         switch(choice)
158         {
159             case 1:
160                 Insert();
161                 break;
162
163             case 2:
164                 deletebeg();
165                 break;
166
167             case 3:
168                 deleteend();
169                 break;
170
171             case 4:
172                 deletemid();
173                 break;
174
175             case 5:
176                 display();
177                 break;
178
179             case 6:
180                 break;
181
182             default:
183                 printf("Wrong choice!\n");
184                 break;
185         }
186     }while(choice!=6);
187     return 0;
188 }
[Finished in 0.3s]
Line 1, Column 5 Spaces: 4 C
```

```
input

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit

Enter your choice : 1
Enter your name : aaaa
Enter your semester : 3
Enter your usn : 1234
first node of linked list created

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit

Enter your choice : 1
Enter your name : ssss
Enter your semester : 3
Enter your usn : 2345
Node created
```

```
input

Enter your choice : 1
Enter your name : ssss
Enter your semester : 3
Enter your usn : 2345
Node created

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit

Enter your choice : 1
Enter your name : ddddd
Enter your semester : 3
Enter your usn : 4567
Node created

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit
```

```
5. display list
6.exit

Enter your choice : 1
Enter your name : ggggg
Enter your semester : 7890
Enter your usn : 7890
Node created

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit

Enter your choice : 2

Node deleted from the begining ...
1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit
```

```
input
Enter your choice : 5
----NODE 1----
Name: ssss
USN: 2345
Sem: 3

----NODE 2----
Name: ddddd
USN: 4567
Sem: 3

----NODE 3----
Name: ggggg
USN: 7890
Sem: 7890

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
4.delete a given node
5. display list
6.exit

Enter your choice : 3
```

Enter your choice : 3

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
- 4.delete a given node
5. display list
- 6.exit

Enter your choice : 5

----NODE 1----

Name: ssss

USN: 2345

Sem: 3

----NODE 2----

Name: ddddd

USN: 4567

Sem: 3

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
- 4.delete a given node
5. display list

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
- 4.delete a given node
5. display list
- 6.exit

Enter your choice : 4

enter the usn of student to be deleted

7890

student not in the list

1. Insert node
2. delete node in the beg of the list
3. delete at the end of list
- 4.delete a given node
5. display list
- 6.exit

Enter your choice : 6

...Program finished with exit code 0

Press ENTER to exit console.

```
age1 Generics.java x del.c a1.c x dept1.css x
1  #include <stdlib.h>
2  #include <string.h>
3  struct node
4  {
5      int sem;
6      struct node *next;
7  };
8  struct node *head= NULL;
9  struct node *head2= NULL;
10 int c=0;
11 void Insert()
12 {
13     struct node *newnode;
14     struct node *temp;
15     int s;
16     printf("Enter integer : ");
17     scanf("%d",&s);
18     newnode=(struct node*)malloc(sizeof(struct node));
19     newnode->sem =s;
20     if (head==NULL)
21     {
22         newnode->next=NULL;
23         head=newnode;
24         printf("first node of linked list created\n");
25         c++;
26     }
27     else
28     {
29         temp=head;
30         while(temp->next!=NULL)
31         {
32             temp=temp->next;
33         }
34         temp->next=newnode;
35         newnode->next=NULL;
36         c++;
37         printf("Node created\n");
38     }
39 }
```

```
age1 Generics.java x del.c a1.c x dept1.css x Student.java — Desktop
40 void Insert2()
41 {
42     struct node *newnode;
43     struct node *temp;
44     int s,y;
45     printf("enter elements to create list 2\n");
46     do
47     {
48         printf("Enter integer : \n");
49         scanf("%d",&s);
50         newnode=(struct node*)malloc(sizeof(struct node));
51         newnode->sem =s;
52         if (head2==NULL)
53         {
54             newnode->next=NULL;
55             head2=newnode;
56             printf("first node of linked list created\n");
57             c++;
58         }
59         else
60         {
61             temp=head2;
62             while(temp->next!=NULL)
63             {
64                 temp=temp->next;
65             }
66             temp->next=newnode;
67             newnode->next=NULL;
68             c++;
69             printf("Node created\n");
70         }
71         printf("do u want to continue adding:0 or 1\n");
72         scanf("%d",&y);
73     }while(y!=0);
74 }
75
```

```

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

```

```

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

```





```
195     switch(choice)
196     {
197         case 1:
198             Insert();
199             break;
200
201         case 2:
202             printf("before:\n");
203             display1();
204             bubbleSort();
205             printf("after:\n");
206             display1();
207             break;
208
209         case 3:
210             printf("before:\n");
211             display1();
212             reverse();
213             printf("after:\n");
214
215             display1();
216             break;
217
218         case 4:
219             Insert2();
220             concat();
221             display1();
222             break;
223
224         case 5:
225             break;
226
227         default:
228             printf("Wrong choice!\n");
229             break;
230     }
231 }while(choice!=5);
232 return 0;
233 }
```

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

Enter your choice : 1  
Enter integer : 1  
first node of linked list created

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

Enter your choice : 1  
Enter integer : 4  
Node created

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

Enter your choice : 1

Enter integer : 6

Node created

1. Insert node

2. sort node

3. reverse node

4.concat 2 lists

5.exit

Enter your choice : 1

Enter integer : 3

Node created

1. Insert node

2. sort node

3. reverse node

4.concat 2 lists

5.exit

Enter your choice : 2

before:

1 4 6 3

after:

1 3 4 6

1. Insert node

2. sort node

3. reverse node

4.concat 2 lists

5.exit

Enter your choice : 3

before:

1 3 4 6

after:

6 4 3 1

1. Insert node

2. sort node

3. reverse node

4.concat 2 lists

5.exit

Enter your choice : 4

before:

6 4 3 1 enter elements to create list 2

Enter integer :

2

first node of linked list created

do u want to continue adding:0 or 1

1

```
do u want to continue adding:0 or 1
1
Enter integer  :
4
Node created

do u want to continue adding:0 or 1
0

after:
 6 4 3 1 2 4
1. Insert node
2. sort node
3. reverse node
4.concat 2 lists
5.exit

Enter your choice : 5

...Program finished with exit code 0
Press ENTER to exit console.
```