

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > *o NODE
1 #include <stdio.h>
2 #include <stdlib.h>
3 int count = 0;
4 struct Node
5 {
6     int item;
7     struct Node *link;
8 };
9 typedef struct Node *NODE;
10 NODE getNode()
11 {
12     NODE temp;
13     temp = (NODE)malloc(sizeof(struct Node));
14     return temp;
15 }
16 void freeNode(NODE x)
17 {
18     free(x);
19 }
20 NODE insert_front(NODE first, int item)
21 {
22     NODE temp = getNode();
23     count += 1;
24     temp->item = item;
25     temp->link = NULL;
26     if (first == NULL)
27         return temp;
28     temp->link = first;
29     first = temp;
30     return first;
31 }
32 NODE insert_rear(NODE first, int item)
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > *o NODE
32 NODE insert_rear(NODE first, int item)
33 {
34     NODE temp, x;
35     count += 1;
36     temp = getNode();
37     temp->item = item;
38     temp->link = NULL;
39     x = first;
40     if (x == NULL)
41     {
42         return temp;
43     }
44     while (x->link != NULL)
45     {
46         x = x->link;
47     }
48     x->link = temp;
49     return first;
50 }
51
52 NODE delete_rear(NODE first)
53 {
54     NODE cur, prev;
55     count -= 1;
56     if (first == NULL)
57     {
58         printf("List is empty");
59         return first;
60     }
61     if (first->link == NULL)
62     {
63         printf("item deleted is %d\n", first->item);
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > search(NODE, int)
63     printf("item deleted is %d\n", first->item);
64     free(first);
65     return NULL;
66 }
67 prev = NULL;
68 cur = first;
69 while (cur->link != NULL)
70 {
71     prev = cur;
72     cur = cur->link;
73 }
74 printf("Item deleted at rearend is %d\n", cur->item);
75 free(cur);
76 prev->link = NULL;
77 return first;
78 }
79 NODE deletefront(NODE first)
80 {
81     NODE temp;
82     count -= 1;
83     temp = first;
84     temp = temp->link;
85     free(first);
86     first = temp;
87     return first;
88 }
89 NODE search(NODE first, int key)
90 {
91     if (first->link == NULL)
92     {
93         printf("List is empty\n");
94         return first;
95     }
96     NODE temp;
97     temp = first;
98     int count=0;
99     if (first==NULL)
100     printf("List is empty\n");
101     while (temp!= NULL)
102     {
103         count++;
104         if (temp->item == key)
105         {
106             printf("Key found\n",count);
107             return first;
108         }
109         temp=temp->link;
110     }
111     printf("Key not found\n");
112     return first;
113 }
114 void display(NODE first)
115 {
116     NODE temp;
117     if (first == NULL)
118     printf("list empty cannot display items\n");
119     for (temp = first; temp != NULL; temp = temp->link)
120     {
121         printf("%d\n", temp->item);
122     }
123 }
124 NODE order(NODE first, int item)
125 {
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > display(NODE)
93     printf("List is empty");
94     return first;
95 }
96 NODE temp;
97 temp = first;
98 int count=0;
99 if (first==NULL)
100 printf("List is empty\n");
101 while (temp!= NULL)
102 {
103     count++;
104     if (temp->item == key)
105     {
106         printf("Key found\n",count);
107         return first;
108     }
109     temp=temp->link;
110 }
111 printf("Key not found\n");
112 return first;
113 }
114 void display(NODE first)
115 {
116     NODE temp;
117     if (first == NULL)
118     printf("list empty cannot display items\n");
119     for (temp = first; temp != NULL; temp = temp->link)
120     {
121         printf("%d\n", temp->item);
122     }
123 }
124 NODE order(NODE first, int item)
125 {
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > reverse(NODE)
123 NODE order(NODE first, int item)
124 {
125     NODE temp, prev, cur;
126     temp = getNode();
127     temp->item = item;
128     temp->link = NULL;
129     if (first == NULL)
130         return temp;
131     if (item < first->item)
132     {
133         temp->link = first;
134         return temp;
135     }
136     prev = NULL;
137     cur = first;
138     while (cur != NULL && item > cur->item)
139     {
140         prev = cur;
141         cur = cur->link;
142     }
143     prev->link = temp;
144     temp->link = cur;
145     return first;
146 }
147 NODE reverse(NODE first)
148 {
149     NODE cur, temp;
150     cur = NULL;
151     while (first != NULL)
152     {
153         temp = first;
154         first = first->link;
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > reverse(NODE)
147 NODE reverse(NODE first)
148 {
149     NODE cur, temp;
150     cur = NULL;
151     while (first != NULL)
152     {
153         temp = first;
154         first = first->link;
155         temp->link = cur;
156         cur = temp;
157     }
158     return cur;
159 }
160 NODE del_key(NODE first, int key)
161 {
162     NODE prev;
163     NODE temp = first;
164     prev = first;
165     while (temp->item != key)
166     {
167         if (temp->link == NULL)
168         {
169             prev->link = NULL;
170             printf("\n No result found \n");
171             return first;
172         }
173         prev = temp;
174         temp = temp->link;
175     }
176     prev->link = temp->link;
177     count -= 1;
178     freeNode(temp);
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > del_key(NODE, int)
179     return first;
180 }
181 NODE ins_key(NODE first, int item, int pos)
182 {
183     NODE prev, cur;
184     cur = first;
185     if (pos > count + 1)
186     {
187         printf("Enter correct position ");
188         return first;
189     }
190     NODE temp = getNode();
191     temp->item = item;
192     temp->link = NULL;
193     int c = 1;
194     prev = cur;
195     if (pos==1)
196         return insert_front(first,item);
197     while (c != pos)
198     {
199         prev = cur;
200         cur = cur->link;
201         c += 1;
202     }
203     temp->link = cur;
204     prev->link = temp;
205     return first;
206 }
207 NODE Sorting(NODE first)
208 {
209     int temp;
210     NODE cur = first;
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > main()
207     NODE Sorting(NODE first)
208     {
209         int temp;
210         NODE cur = first;
211         if (cur->link==NULL){
212             return cur;
213         }
214         NODE index;
215         while(cur != NULL) {
216             index = cur->link;
217             while(index != NULL) {
218                 if(cur->item > index->item) {
219                     temp = cur->item;
220                     cur->item = index->item;
221                     index->item = temp;
222                 }
223                 index = index->link;
224             }
225             cur = cur->link;
226         }
227         return first;
228     }
229 void main()
230 {
231     int item, choice=1, pos;
232     NODE first = NULL;
233     while(choice!=0)
234     {
235         printf("\n 0: Exit \n 1: Insert_front\n 2: Delete_front\n 3: Insert_rear\n 4: Delete_rear\n 5: Display_list\n 6: Order\n 7:
Reverse \n 8: Search \n 9: Delete at specific position \n 10: Insert at any position \n 11: Sorting \n");
236         printf("\n enter the choice\n");
237         scanf("%d", &choice);
```

```
File Edit Selection View Go Run Terminal Help linkedlist - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > main()
229 void main()
230 {
231     int item, choice=1, pos;
232     NODE first = NULL;
233     while(choice!=0)
234     {
235         printf("\n 0: Exit \n 1: Insert_front\n 2: Delete_front\n 3: Insert_rear\n 4: Delete_rear\n 5: Display_list\n 6: Order\n 7:
Reverse \n 8: Search \n 9: Delete at specific position \n 10: Insert at any position \n 11: Sorting \n");
236         printf(" enter the choice\n ");
237         scanf("%d", &choice);
238         switch (choice)
239         {
240             case 0:
241                 exit(0);
242             case 1:
243                 printf("enter the item at front - end\n");
244                 scanf("%d", &item);
245                 first = insert_front(first, item);
246                 break;
247             case 2:
248                 first = deletefront(first);
249                 display(first);
250                 break;
251             case 3:
252                 printf("Enter the item at rear end");
253                 scanf("%d", &item);
254                 first = insert_rear(first, item);
255                 break;
256             case 4:
257                 first = delete_rear(first);
258                 display(first);
259                 break;
```

```
File Edit Selection View Go Run Terminal Help linkedlist - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > main()
260         case 5:
261             display(first);
262             break;
263         case 6:
264             printf("enter the item\n");
265             scanf("%d", &item);
266             first = order(first, item);
267             display(first);
268             break;
269         case 7:
270             first = reverse(first);
271             display(first);
272             break;
273         case 8:
274             printf("Enter the item to searched for\n");
275             scanf("%d", &item);
276             first=search(first, item);
277             break;
278         case 9:
279             printf("Deleted A specific key");
280             scanf("%d", &item);
281             first = del_key(first, item);
282             display(first);
283             break;
284         case 10:
285             printf("Enter the position of item to be inserted");
286             scanf("%d", &pos);
287             printf("Enter item to be inseted at any position");
288             scanf("%d", &item);
289             first = ins_key(first, item, pos);
290             display(first);
291             break;
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code

C linkedlist.c X
linkedlist > C linkedlist.c > main()

271     display(first);
272     break;
273
274     case 8:
275         printf("Enter the item to searched for\n");
276         scanf("%d", &item);
277         first=search(first, item);
278         break;
279
280     case 9:
281         printf("Deleted A specific key");
282         scanf("%d", &item);
283         first = del_key(first, item);
284         display(first);
285         break;
286
287     case 10:
288         printf("Enter the position of item to be inserted");
289         scanf("%d", &pos);
290         printf("Enter item to be inseted at any position");
291         scanf("%d", &item);
292         first = ins_key(first, item, pos);
293         display(first);
294         break;
295
296     case 11:
297         printf("\n Sorting the list \n");
298         first=Sorting(first);
299         display(first);
300         break;
301     default:
302         exit(0);
303     }
304 }
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\c\college\classwork> cd linkedlist
PS D:\c\college\classwork\linkedlist> gcc linkedlist.c
PS D:\c\college\classwork\linkedlist> ./a.exe

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
1
enter the item at front - end
10

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell
enter the choice
1
enter the item at front - end
20

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
1
enter the item at front - end
30

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
3
Enter the item at rear end40

0: Exit
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell
enter the choice
1
enter the item at front - end
20

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
1
enter the item at front - end
30

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
3
Enter the item at rear end40

0: Exit
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
2
20
10
40
50

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
4
Item deleted at rearend is 50
20
10
40
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
3
Enter the item at rear end40

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
3
Enter the item at rear end30

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
```



```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell

5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
5
20
10
40
40
30

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
5
20
10
40
40
30

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
6
enter the item
15
15
20
10
40
40
30

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
6
enter the item
25
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell
enter the item
25
15
20
10
25
40
40
30
0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
7
30
40
40
25
10
20
15
0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
9
Deleted A specific key40
30
40
25
10
20
15
0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
10
Enter the position of item to be inserted4
Enter item to be inseted at any position70
30
40
25
70
10
20
15
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell
0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
11

Sorting the list
10
15
20
25
30
40
70

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
8
Enter the item to searched for
```

```
File Edit Selection View Go Run Terminal Help linkedlist.c - classwork - Visual Studio Code
1: powershell
20
25
30
40
70

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
8
Enter the item to searched for
30
Key found

0: Exit
1: Insert_front
2: Delete_front
3: Insert_rear
4: Delete_rear
5: Display_list
6: Order
7: Reverse
8: Search
9: Delete at specific position
10: Insert at any position
11: Sorting
enter the choice
0
PS D:\c\college\classwork\linkedList>
```