

create a singly linked list to create a linked list, delete front, rear, from a particular position

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

```
main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct node{
4     int info;
5     struct node *link;
6 };
7 typedef struct node *NODE;
8 NODE getnode()
9 {
10     NODE x;
11     x=(NODE)malloc(sizeof(NODE));//
12     if(x==NULL)
13     {
14         printf("memory full \n");
15         exit(0);
16     }
17     return x;
18 }
19
20 void freenode(NODE x)
21 {
22     free(x);
23 }
24
25 NODE insert_front(NODE first,int item)
26 {
27     NODE temp = getnode();
28     temp->info = item;
29     temp->link = NULL;
```

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

```
main.c
23 }
24
25 NODE insert_front(NODE first,int item)
26 {
27     NODE temp = getnode();
28     temp->info = item;
29     temp->link = NULL;
30     if(first == NULL)
31         return temp;
32     temp->link=first;
33     return temp;
34 }
35
36 NODE delete_front(NODE first)
37 {
38     NODE temp;
39     if(first == NULL)
40     {
41         printf("List is empty\n");
42         return first;
43     }
44     printf("Item deleted %d",(first->info));
45     temp = first;
46     temp=temp->link;
47     free(first);
48     return temp;
49 }
50
51 NODE insert_rear(NODE first,int item)
```

onlinegdb.com/online_c_compiler

☆ 🔍 🔧 ⓘ ⚙️

Run Debug Stop Share Save {} Beautify

Language C

main.c

```
50 }
51 NODE insert_rear(NODE first,int item)
52 {
53     NODE temp = getnode(),cur;
54     temp->info=item;
55     temp->link = NULL;
56     if(first==NULL)
57         return temp;
58     cur=first;
59     while(cur->link!=NULL)
60         cur=cur->link;
61     cur->link=temp;
62     return first;
63 }
64
65 NODE delete_rear(NODE first)
66 {
67     NODE cur=first,prev=NULL;
68     if(first==NULL)
69     {
70         printf("List empty\n");
71         return NULL;
72     }
73     if(first->link==NULL)
74     {
75         printf("item deleted is %d\n",first->info);
76         free(first);
77         return NULL;
```

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

main.c

```
73 if(first->link==NULL)
74 {
75     printf("item deleted is %d\n",first->info);
76     free(first);
77     return NULL;
78 }
79 while(cur->link!=NULL)
80 {
81     prev=cur;
82     cur=cur->link;
83 }
84 printf("Item deleted is %d\n",(cur->info));
85 free(cur);
86 prev->link=NULL;
87 return first;
88 }
89
90 void display(NODE first)
91 {
92     if(first==NULL)
93     {
94         printf("List is empty\n");
95         return;
96     }
97     printf("Elements of the list are : \n");
98     for(NODE i=first;i!=NULL;i=i->link)
99         printf("%d\n",i->info);
100 }
101 NODE insert_pos(NODE first,int item,int pos)
```

```
onlinegdb.com/online_c_compiler
Language C
main.c
97 printf("Elements of the list are : \n");
98 for(NODE i=first;i!=NULL;i=i->link)
99 printf("%d\n",i->info);
100 }
101 NODE insert_pos(NODE first,int item,int pos)
102 {
103     int c=1;
104     NODE temp = getnode(),cur,prev;
105     temp->info=item;
106     if(pos==1)
107     {
108         temp->link=first;
109         return temp;
110     }
111     cur=first;
112     prev=NULL;
113     while(cur!=NULL)
114     {
115         if(pos==c)
116         {
117             prev->link=temp;
118             temp->link=cur;
119             return first;
120         }
121         c++;
122         prev=cur;
123         cur=cur->link;
124     }
125     printf("Invalid position\n");
126 }
```

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

```
main.c
120     }
121     c++;
122     prev=cur;
123     cur=cur->link;
124 }
125 printf("Invalid position\n");
126 return first;
127 }
128
129 NODE delete_pos(NODE first,int pos)
130 {
131     NODE cur,prev;
132     int c=1;
133     if(first==NULL||pos<0)
134     {
135         printf("Invalid position\n");
136         return NULL;
137     }
138     if(pos==1)
139     {
140         free(first);
141         return NULL;
142     }
143     cur=first;
144     prev=NULL;
145     while(cur!=NULL)
146     {
147         if(c==pos)
148         {
```

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

```
main.c
140 free(first);
141 return NULL;
142 }
143 cur=first;
144 prev=NULL;
145 while(cur!=NULL)
146 {
147     if(c==pos)
148     {
149         printf("Element deleted is %d",cur->info);
150         prev->link=cur->link;
151         free(cur);
152         return first;
153     }
154     prev=cur;
155     cur=cur->link;
156     c++;
157 }
158 printf("Element not found\n");
159 return first;
160 }
161 NODE delete_key(NODE first,int key)
162 {
163     NODE prev,cur;
164     if(first==NULL)
165     {
166         printf("List is empty\n");
167         return NULL;
168     }
169 }
```


onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

main.c

```
165 {
166     printf("List is empty\n");
167     return NULL;
168 }
169 if(key==first->info)
170 {
171     cur=first;
172     first=first->link;
173     free(cur);
174     printf("Element deleted successfully\n");
175     return first;
176 }
177 cur=first;
178 prev=NULL;
179 while(cur!=NULL)
180 {
181     if(key==cur->info)
182     {
183         printf("Item deleted successfully\n");
184         prev->link=cur->link;
185         free(cur);
186         return first;
187     }
188     prev=cur;
189     cur=cur->link;
190 }
191 if(cur==NULL)
192     printf("Element not found \n");
193 }
```

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

```
main.c
179 while(cur!=NULL)
180 {
181     if(key==cur->info)
182     {
183         printf("Item deleted successfully\n");
184         prev->link=cur->link;
185         free(cur);
186         return first;
187     }
188     prev=cur;
189     cur=cur->link;
190 }
191 if(cur==NULL)
192     printf("Element not found \n");
193     return first;
194 }
195
196 int main()
197 {
198     int item,ch,pos;
199     NODE first=NULL;
200     for(;;)
201     {
202         printf("\n1.Insert front\n2.Delete front\n3.Insert rear\n4.delete rear\n5.Insert_pos\n6.Delete_pos\n7.Delete key\n8.Display\n");
203         scanf("%d",&ch);
204         switch(ch)
205         {
206             case 1:
207                 printf("Enter element to be inserted\n");
208
```

```
onlinegdb.com/online_c_compiler
Run Debug Stop Share Save {} Beautify
Language C
main.c
202 printf("\n1.Insert front\n2.Delete front\n3.Insert rear\n4.delete rear\n5.Insert_pos\n6.Delete_pos\n7.Delete key\n8.Display\n");
203 scanf("%d",&ch);
204 switch(ch)
205 {
206     case 1:
207         printf("Enter element to be inserted\n");
208         scanf("%d",&item);
209         first = insert_front(first,item);
210         break;
211     case 2:
212         first = delete_front(first);
213         break;
214     case 3:
215         printf("Enter element to be inserted\n");
216         scanf("%d",&item);
217         first = insert_rear(first,item);
218         break;
219     case 4:
220         first = delete_rear(first);
221         break;
222     case 5:
223         printf("Enter element to be inserted\n");
224         scanf("%d",&item);
225         printf("Enter position\n");
226         scanf("%d",&pos);
227         first = insert_pos(first,item,pos);
228         break;
229     case 6:
230         printf("Enter position\n");
231
```

onlinegdb.com/online_c_compiler

Run Debug Stop Share Save {} Beautify

Language C

```
main.c
217     first = insert_rear(first,item);
218     break;
219 case 4:
220     first = delete_rear(first);
221     break;
222 case 5:
223     printf("Enter element to be inserted\n");
224     scanf("%d",&item);
225     printf("Enter position\n");
226     scanf("%d",&pos);
227     first = insert_pos(first,item,pos);
228     break;
229 case 6:
230     printf("Enter position\n");
231     scanf("%d",&pos);
232     first = delete_pos(first,pos);
233     break;
234 case 7:
235     printf("Enter element to be deleted\n");
236     scanf("%d",&item);
237     first = delete_key(first,item);
238     break;
239 case 8:
240     display(first);
241     break;
242 default: return 0;
243 }
244 }
245 }
```

```
input
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit

Enter element to be inserted
)

Insert front
Delete front
Insert rear
delete rear
Insert_pos
Delete_pos
Delete key
Display
Exit

er element to be inserted

nsert front
```

input

```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
```

```
1
Enter element to be inserted
30
```

```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
```

```
1
Enter element to be inserted
40
```

```
1.Insert front
2.Delete front
3.Insert rear
```





input

```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
```

```
Enter element to be inserted
)
```

```
Insert front
Delete front
Insert rear
delete rear
Insert_pos
Delete_pos
Delete key
Display
Exit
```

```
Enter element to be inserted
```

```
Insert front
```



```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
4
Item deleted is 10
```

```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
8
Elements of the list are :
40
30
20
```

```
1.Insert front
```


20
10

1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
4
Item deleted is 10

1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
8
Elements of the list are :
40
30
20

```
7
Enter element to be deleted
20
Item deleted successfully
```

```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
8
Elements of the list are :
60
40
```

```
1.Insert front
2.Delete front
3.Insert rear
4.delete rear
5.Insert_pos
6.Delete_pos
7.Delete key
8.Display
9.Exit
```

