

 **OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new


Learn Programming

Programming Questions

Sign Up

Login

   5.1K

 **Red Hat**

Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2020 GDB Online

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 NODE x;
10 x=(NODE)malloc(sizeof(struct node));
11 if(x==NULL){
12 printf("Memory full\n");
13 exit(0);
14 }
15 return x;
16 }
17 void freenode(NODE x){
18 free(x);
19 }
20 NODE insert_front(NODE first,int item){
21 NODE temp;
22 temp=getnode();
23 temp->info=item;
24 temp->link=NULL;
25 if(first==NULL)
26 return temp;
27 temp->link=first;
28 first=temp;
29 return first;
30 }
31 NODE delete_front(NODE first){
32 NODE temp;
33 if(first==NULL){
34 printf("List is empty cannot delete\n");
35 return first;
36 }
```

**OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new


Learn Programming

Programming Questions

Sign Up

Login





Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2020 GDB Online

```
main.c
35 return first;
36 }
37 temp=first;
38 temp=temp->link;
39 printf("Item deleted at front end is %d\n",first->info);
40 free(first);
41 return temp;
42 }
43 NODE insert_rear(NODE first,int item){
44 NODE temp,cur;
45 temp=getnode();
46 temp->info=item;
47 temp->link=NULL;
48 if(first==NULL)
49 return temp;
50 cur=first;
51 while(cur->link!=NULL)
52 cur=cur->link;
53 cur->link=temp;
54 return first;
55 }
56 NODE delete_rear(NODE first){
57 NODE cur,prev;
58 if(first==NULL){
59 printf("List is empty cannot delete\n");
60 return first;
61 }
62 if(first->link==NULL){
63 printf("Item deleted is %d\n",first->info);
64 free(first);
65 return NULL;
66 }
67 prev=NULL;
68 cur=first;
69 while(cur->link!=NULL){
70
```

**OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new


Learn Programming

Programming Questions

Sign Up

Login

 5.1K


**Red Hat**

Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

```
main.c
69 - while(cur->link!=NULL){
70     prev=cur;
71     cur=cur->link;
72 }
73 printf("Item deleted at rear end is %d",cur->info);
74 free(cur);
75 prev->link=NULL;
76 return first;
77 }
78 - NODE insert_pos(int item,int pos,NODE first){
79     NODE temp,cur,prev;
80     int count;
81     temp=getnode();
82     temp->info=item;
83     temp->link=NULL;
84 - if(first==NULL&&pos==1){
85     return temp;
86 }
87 - if(first==NULL){
88     printf("Invalid position\n");
89     return first;
90 }
91 - if(pos==1){
92     temp->link=first;
93     first=temp;
94     return temp;
95 }
96 count=1;
97 prev=NULL;
98 cur=first;
99 - while(cur!=NULL&&count!=pos){
100     prev=cur;
101     cur=cur->link;
102     count++;
103 }
104 -
```



**OnlineGDB** beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE

My Projects




Classroom new


Learn Programming

Programming Questions

Sign Up

Login

 5.1K


**Red Hat**

Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy

© 2016 - 2020 GDB Online



Language C

main.c

```
103 }
104 if(count==pos){
105 prev->link=temp;
106 temp->link=cur;
107 return first;
108 }
109 printf("Invalid position\n");
110 return first;
111 }
112 NODE delete_pos(int pos,NODE first){
113 NODE cur;
114 NODE prev;
115 int count,flag=0;
116 if(first==NULL || pos<0){
117 printf("Invalid position\n");
118 return NULL;
119 }
120 if(pos==1){
121 cur=first;
122 first=first->link;
123 freenode(cur);
124 return first;
125 }
126 prev=NULL;
127 cur=first;
128 count=1;
129 while(cur!=NULL){
130 if(count==pos){
131 flag=1;
132 break;
133 }
134 count++;
135 prev=cur;
136 cur=cur->link;
137 }
138 }
```



 **OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

[IDE](#)

[My Projects](#)

[Classroom new](#)


[Learn Programming](#)

[Programming Questions](#)

[Sign Up](#)

[Login](#)



Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

```
137 }
138 if(flag==0){
139     printf("Invalid position\n");
140     return first;
141 }
142 printf("Item deleted at given position is %d\n",cur->info);
143 prev->link=cur->link;
144 freenode(cur);
145 return first;
146 }
147 void display(NODE first){
148     NODE temp;
149     if(first==NULL)
150         printf("List empty cannot display items\n");
151     for(temp=first;temp!=NULL;temp=temp->link){
152         printf("%d\t",temp->info);
153     }
154 }
155 void main()
156 {
157     int item,choice,key,pos;
158     int count=0;
159     NODE first=NULL;
160     for(;;){
161         printf("\n1:Insert rear\n2:Delete rear\n3:Insert front\n4:Delete front\n5:Insert info position\n6:Delete info position\n7:Display\n");
162         printf("Enter the choice: ");
163         scanf("%d",&choice);
164         switch(choice){
165             case 1:printf("Enter the item at rear end\n");
166                 scanf("%d",&item);
167                 first=insert_rear(first,item);
168                 break;
169             case 2:first=delete_rear(first);
170                 break;
171             case 3:printf("Enter the item at front end\n");
172                 
```

**OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

[IDE](#)

[My Projects](#)

[Classroom new](#)


[Learn Programming](#)

[Programming Questions](#)

[Sign Up](#)

[Login](#)

 5.1K



Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2020 GDB Online

```
159 NODE first=NULL;
160 for(;;){
161     printf("\n1:Insert rear\n2:Delete rear\n3:Insert front\n4:Delete front\n5:Insert info position\n6:Delete info position\n7:Display :
162     printf("Enter the choice: ");
163     scanf("%d",&choice);
164     switch(choice){
165     case 1:printf("Enter the item at rear end\n");
166     scanf("%d",&item);
167     first=insert_rear(first,item);
168     break;
169     case 2:first=delete_rear(first);
170     break;
171     case 3:printf("Enter the item at front end\n");|
172     scanf("%d",&item);
173     first=insert_front(first,item);
174     break;
175     case 4:first=delete_front(first);
176     break;
177     case 5:printf("Enter the item to be inserted at given position\n");
178     scanf("%d",&item);
179     printf("Enter the position\n");
180     scanf("%d",&pos);
181     first=insert_pos(item,pos,first);
182     break;
183     case 6:printf("Enter the position\n");
184     scanf("%d",&pos);
185     first=delete_pos(pos,first);
186     break;
187     case 7:display(first);
188     break;
189     default:exit(0);
190     break;
191     }
192 }
193 }
```

OnlineGDB

beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Sign Up

Login

f

t

5.1K

Red Hat

Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy

© 2016 - 2020 GDB Online

input

1:Insert rear

2:Delete rear

3:Insert front

4:Delete front

5:Insert info position

6:Delete info position

7:Display list

8:Exit

Enter the choice: 3

Enter the item at front end

1

1:Insert rear

2:Delete rear

3:Insert front

4:Delete front

5:Insert info position

6:Delete info position

7:Display list

8:Exit

Enter the choice: 3

Enter the item at front end

2

1:Insert rear

2:Delete rear

3:Insert front

4:Delete front

5:Insert info position

6:Delete info position

7:Display list


8:Exit

Enter the choice: 3

**OnlineGDB** beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

[IDE](#)
[My Projects](#)
[Classroom](#) new
[Learn Programming](#)
[Programming Questions](#)
[Sign Up](#)
[Login](#)

 5.1K


Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.
ADS VIA CARBON

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)
© 2016 - 2020 GDB Online

input

```
Enter the choice: 3
Enter the item at front end
3
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 1
Enter the item at rear end
4
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 1
Enter the item at rear end
5
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
```


**OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

[IDE](#)

[My Projects](#)

[Classroom](#) new


[Learn Programming](#)

[Programming Questions](#)

[Sign Up](#)

[Login](#)





Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

input

```
6:Delete info position
7:Display list
8:Exit
Enter the choice: 1
Enter the item at rear end
6
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 5
Enter the item to be inserted at given position
7
< Enter the position
3
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 5
Enter the item to be inserted at given position
8
Enter the position
6
```

**OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

[IDE](#)

[My Projects](#)

[Classroom new](#)


[Learn Programming](#)

[Programming Questions](#)

[Sign Up](#)

[Login](#)





Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)

© 2016 - 2020 GDB Online

input

```
6
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 7
3      2      7      1      4      8      5      6
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 2
Item deleted at rear end is 6
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 4
Item deleted at front end is 3
1:Insert rear
```

**OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

[IDE](#)

[My Projects](#)

[Classroom](#) new


[Learn Programming](#)

[Programming Questions](#)

[Sign Up](#)

[Login](#)

 5.1K



Massive demand calls for new approaches. Embrace cloud-native dev with Red Hat.

ADS VIA CARBON

input

```
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 6
Enter the position
5
Item deleted at given position is 8
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 7
2      7      1      4      5
1:Insert rear
2:Delete rear
3:Insert front
4:Delete front
5:Insert info position
6:Delete info position
7:Display list
8:Exit
Enter the choice: 8
...Program finished with exit code 0
Press ENTER to exit console.
```