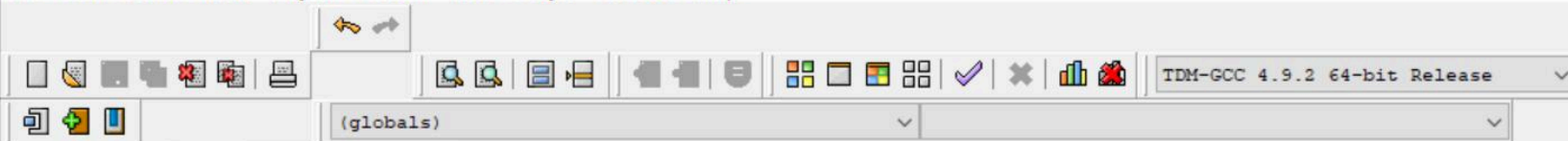


C:\Users\Nikita\Desktop\programs\Singly linked list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help



Project Classes

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int info;
6      struct node *link;
7  };
8  typedef struct node *NODE;
9  NODE getnode()
10 {
11     NODE x;
12     x=(NODE)malloc(sizeof(struct node));
13     if(x==NULL)
14     {
15         printf("memory full\n");
16         exit(0);
17     }
18     return x;
19 }
20 void freenode(NODE x)
21 {
22     free(x);
23 }
24 NODE insert_front(NODE first,int item)
25 {
26     NODE temp;
27     temp=getnode();
28     temp->info=item;
29     temp->link=NULL;
30     if(first==NULL)
31         return temp;
32     temp->link=first;
33     first=temp;
34     return first;
35 }
36
37 NODE insert_rear(NODE first,int item)
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds



Project Classes

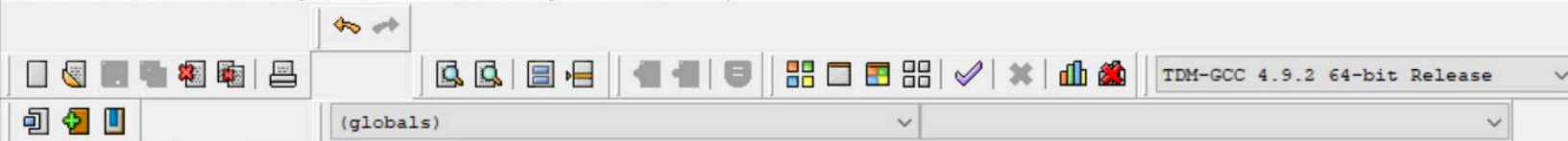
Singly linked list.cpp

```
36
37 NODE insert_rear(NODE first,int item)
38 {
39     NODE temp,cur;
40     temp=getnode();
41     temp->info=item;
42     temp->link=NULL;
43     if(first==NULL)
44         return temp;
45     cur=first;
46     while(cur->link!=NULL)
47         cur=cur->link;
48     cur->link=temp;
49     return first;
50 }
51
52 NODE insert_pos(int item,int pos,NODE first)
53 {
54     NODE temp,cur,prev;
55     int count;
56     temp=getnode();
57     temp->info=item;
58     temp->link=NULL;
59     if (first==NULL && pos==1)
60     {
61         return temp;
62     }
63     if (first==NULL)
64     {
65         printf("Invalid position\n");
66         return NULL;
67     }
68     if (pos==1)
69     {
70         temp->link=first;
71         return temp;
72     }
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds

C:\Users\Nikita\Desktop\programs\Singly linked list.cpp - Dev-C++ 5.11

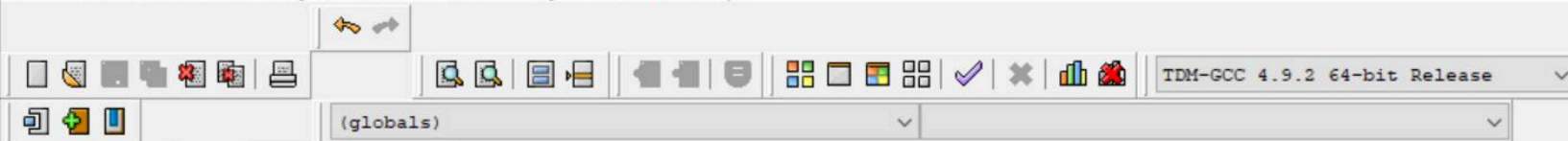
File Edit Search View Project Execute Tools AStyle Window Help



Project Classes

```
60 {
61     return temp;
62 }
63 if (first==NULL)
64 {
65     printf("Invalid position\n");
66     return NULL;
67 }
68 if (pos==1)
69 {
70     temp->link=first;
71     return temp;
72 }
73 count=1;
74 prev=NULL;
75 cur=first;
76 while (cur!=NULL && count!=pos)
77 {
78     prev=cur;
79     cur=cur->link;
80     count++;
81 }
82 if (count==pos)
83 {
84     prev->link=temp;
85     temp->link=cur;
86     return first;
87 }
88 printf("Invalid position\n");
89 return first;
90 }
91
92
93 NODE delete_front(NODE first)
94 {
95     NODE temp;
96     if(first==NULL)
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds



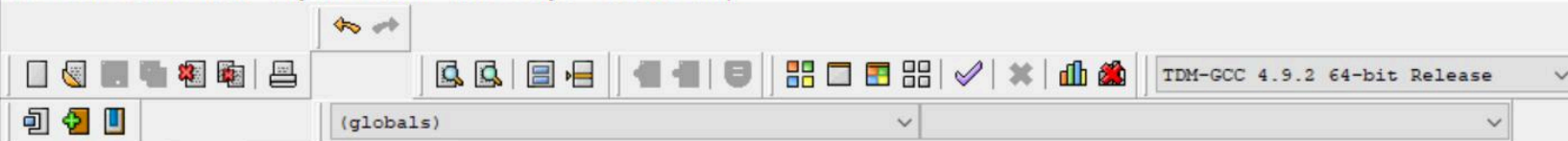
Project Classes

```
93  NODE delete_front(NODE first)
94  {
95      NODE temp;
96      if(first==NULL)
97      {
98          printf("list is empty cannot delete\n");
99          return first;
100     }
101     temp=first;
102     temp=temp->link;
103     printf("item deleted at front-end is=%d\n",first->info);
104     free(first);
105     return temp;
106 }
107
108 NODE delete_rear(NODE first)
109 {
110     NODE cur,prev;
111     if(first==NULL)
112     {
113         printf("list is empty cannot delete\n");
114         return first;
115     }
116     if(first->link==NULL)
117     {
118         printf("item deleted is %d\n",first->info);
119         free(first);
120         return NULL;
121     }
122     prev=NULL;
123     cur=first;
124     while(cur->link!=NULL)
125     {
126         prev=cur;
127         cur=cur->link;
128     }
129     printf("item deleted at rear-end is %d",cur->info);
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds

C:\Users\Nikita\Desktop\programs\Singly linked list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help



Project Classes

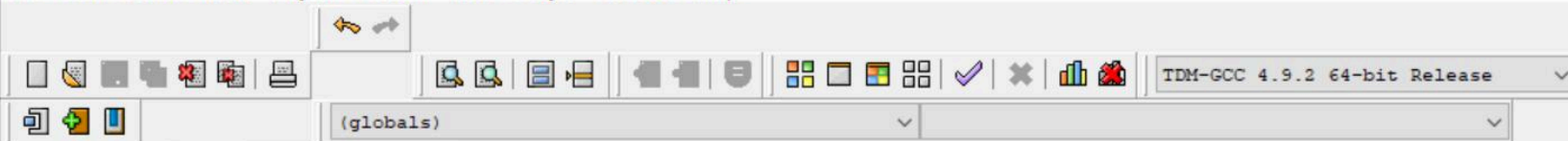
```
114     return first;
115 }
116 if(first->link==NULL)
117 {
118     printf("item deleted is %d\n",first->info);
119     free(first);
120     return NULL;
121 }
122 prev=NULL;
123 cur=first;
124 while(cur->link!=NULL)
125 {
126     prev=cur;
127     cur=cur->link;
128 }
129 printf("item deleted at rear-end is %d",cur->info);
130 free(cur);
131 prev->link=NULL;
132 return first;
133 }
134 NODE delete_pos(int pos,NODE first)
135 {
136     NODE prev,cur;
137     int count;
138     if (first==NULL || pos<=0)
139     {
140         printf("Invalid position\n");
141         return NULL;
142     }
143     if (pos==1)
144     {
145         cur=first;
146         first=first->link;
147         freenode(cur);
148         return first;
149     }
150     prev=NULL;
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds

Type here to search





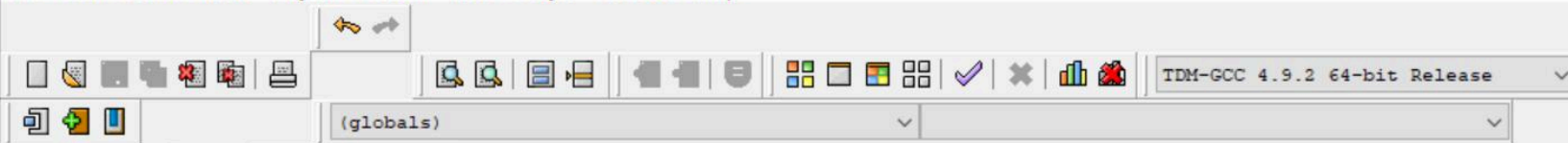


Project Classes

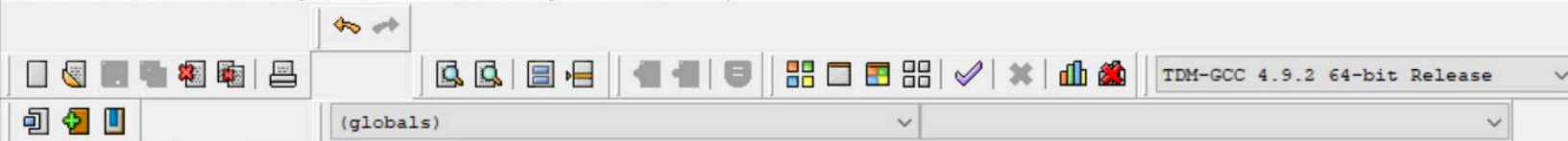
Singly linked list.cpp

```
144 {
145     cur=first;
146     first=first->link;
147     freenode(cur);
148     return first;
149 }
150 prev=NULL;
151 cur=first;
152 count=1;
153 while (cur!=NULL)
154 {
155     if (count==pos)
156     {
157         break;
158     }
159     prev=cur;
160     cur=cur->link;count++;
161 }
162 if (count!=pos)
163 {
164     printf("Invalid position\n");
165     return first;
166 }
167 prev->link=cur->link;
168 freenode(cur);
169 return first;
170 }
171
172 void display(NODE first)
173 {
174     NODE temp;
175     if(first==NULL)
176     printf("list empty cannot display items\n");
177     for(temp=first;temp!=NULL;temp=temp->link)
178     {
179         printf("%d\n",temp->info);
180     }
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds



```
171
172 void display(NODE first)
173 {
174     NODE temp;
175     if(first==NULL)
176         printf("list empty cannot display items\n");
177     for(temp=first;temp!=NULL;temp=temp->link)
178     {
179         printf("%d\n",temp->info);
180     }
181 }
182 int main()
183 {
184     int item,choice,pos;
185     NODE first=NULL;
186
187     for(;;)
188     {
189         printf("\n1:Insert_front\n2:Insert_rear\n3:Insert Position\n4:Delete_front\n5:Delete_rear\n6:Delete Position\n7:display_list\n8:Exit\n");
190         printf("enter the choice\n");
191         scanf("%d",&choice);
192         switch(choice)
193         {
194             case 1:printf("enter the item at front-end\n");
195                     scanf("%d",&item);
196                     first=insert_front(first,item);
197                     break;
198             case 2:printf("enter the item at rear-end\n");
199                     scanf("%d",&item);
200                     first=insert_rear(first,item);
201                     break;
202             case 3:printf("Enter the item and the position:\n");
203                     scanf("%d",&item,&pos);
204                     first=insert_pos(item,pos,first);
205                     break;
206             case 4:first=delete_front(first);
207                     break;
```



Project Classes Singly linked list.cpp

```
186
187 for(;;)
188 {
189     printf("\n1:Insert_front\n2:Insert_rear\n3:Insert Position\n4:Delete_front\n5:Delete_rear\n6:Delete Position\n7:display_list\n8:Exit\n");
190     printf("enter the choice\n");
191     scanf("%d",&choice);
192     switch(choice)
193     {
194     case 1:printf("enter the item at front-end\n");
195             scanf("%d",&item);
196             first=insert_front(first,item);
197             break;
198     case 2:printf("enter the item at rear-end\n");
199             scanf("%d",&item);
200             first=insert_rear(first,item);
201             break;
202     case 3:printf("Enter the item and the position:\n");
203             scanf("%d%d",&item,&pos);
204             first=insert_pos(item,pos,first);
205             break;
206     case 4:first=delete_front(first);
207             break;
208     case 5:first=delete_rear(first);
209             break;
210     case 6:printf("Enter the position:\n");
211             scanf("%d",&pos);
212             first=delete_pos(pos,first);
213             break;
214     case 7:display(first);
215             break;
216     default:exit(0);
217             break;
218     }
219 }
220
221 }
222
```

Line: 139 Col: 6 Sel: 0 Lines: 222 Length: 3638 Insert Done parsing in 0.015 seconds



```
1:Insert_front
2:Insert_rear
3:Insert Position
4:Delete_front
5:Delete_rear
6:Delete Position
7:display_list
8:Exit
enter the choice
1
enter the item at front-end
20
```

```
1:Insert_front
2:Insert_rear
3:Insert Position
4:Delete_front
5:Delete_rear
6:Delete Position
7:display_list
8:Exit
enter the choice
1
enter the item at front-end
30
```

```
1:Insert_front
2:Insert_rear
3:Insert Position
4:Delete_front
5:Delete_rear
6:Delete Position
7:display_list
8:Exit
enter the choice
1
enter the item at front-end
40
```

```
1:Insert_front
```

C:\Users\Nikita\Desktop\programs\Singly linked list.exe

1  
enter the item at front-end  
40

1:Insert\_front  
2:Insert\_rear  
3:Insert Position  
4:Delete\_front  
5:Delete\_rear  
6:Delete Position  
7:display\_list  
8:Exit

enter the choice  
1  
enter the item at front-end  
50

1:Insert\_front  
2:Insert\_rear  
3:Insert Position  
4:Delete\_front  
5:Delete\_rear  
6:Delete Position  
7:display\_list  
8:Exit

enter the choice  
2  
enter the item at rear-end  
10

1:Insert\_front  
2:Insert\_rear  
3:Insert Position  
4:Delete\_front  
5:Delete\_rear  
6:Delete Position  
7:display\_list  
8:Exit

enter the choice  
7  
50

C:\Users\Nikita\Desktop\programs\Singly linked list.exe

50

```
1:Insert_front
2:Insert_rear
3:Insert Position
4>Delete_front
5>Delete_rear
6>Delete Position
7:display_list
8:Exit
enter the choice
2
enter the item at rear-end
10
```

```
1:Insert_front
2:Insert_rear
3:Insert Position
4>Delete_front
5>Delete_rear
6>Delete Position
7:display_list
8:Exit
enter the choice
7
50
40
30
20
10
```

```
1:Insert_front
2:Insert_rear
3:Insert Position
4>Delete_front
5>Delete_rear
6>Delete Position
7:display_list
8:Exit
enter the choice
3
```

C:\Users\Nikita\Desktop\programs\Singly linked list.exe

```
3:Insert Position
4:Delete_front
5:Delete_rear
6:Delete Position
7:display_list
8:Exit
enter the choice
3
Enter the item and the position:
35 3
```

```
1:Insert_front
2:Insert_rear
3:Insert Position
4:Delete_front
5:Delete_rear
6:Delete Position
7:display_list
8:Exit
enter the choice
7
50
40
35
30
20
10
```

```
1:Insert_front
2:Insert_rear
3:Insert Position
4:Delete_front
5:Delete_rear
6:Delete Position
7:display_list
8:Exit
enter the choice
4
item deleted at front-end is=50
```

```
1:Insert_front
```

C:\Users\Nikita\Desktop\programs\Singly linked list.exe

8:Exit  
enter the choice  
4  
item deleted at front-end is=50

1:Insert\_front  
2:Insert\_rear  
3:Insert Position  
4:Delete\_front  
5:Delete\_rear  
6:Delete Position  
7:display\_list  
8:Exit  
enter the choice  
5  
item deleted at rear-end is 10

1:Insert\_front  
2:Insert\_rear  
3:Insert Position  
4:Delete\_front  
5:Delete\_rear  
6:Delete Position  
7:display\_list  
8:Exit  
enter the choice  
6  
Enter the position:  
3

1:Insert\_front  
2:Insert\_rear  
3:Insert Position  
4:Delete\_front  
5:Delete\_rear  
6:Delete Position  
7:display\_list  
8:Exit  
enter the choice  
7  
40  
35



C:\Users\Nikita\Desktop\programs\Singly linked list.exe

```
7:display_list
8:Exit
enter the choice
6
Enter the position:
3

1:Insert_front
2:Insert_rear
3:Insert Position
4>Delete_front
5>Delete_rear
6>Delete Position
7:display_list
8:Exit
enter the choice
7
40
35
20

1:Insert_front
2:Insert_rear
3:Insert Position
4>Delete_front
5>Delete_rear
6>Delete Position
7:display_list
8:Exit
enter the choice
```

Type here to search



12:23  
25-11-2020

