

Document Project Build Tools Help

Save All Revert Close Back Forward Compile Build Execute Color Chooser Find Jump to Quit

ack.c conversion.c evaluation.c factorial.c gcd.c queue.c circular queue.c priorityqueue.c ascending.c descending.c linkedlist1.c linkedlist2.c

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

Nitin - Geany

View Document Project Build Tools Help

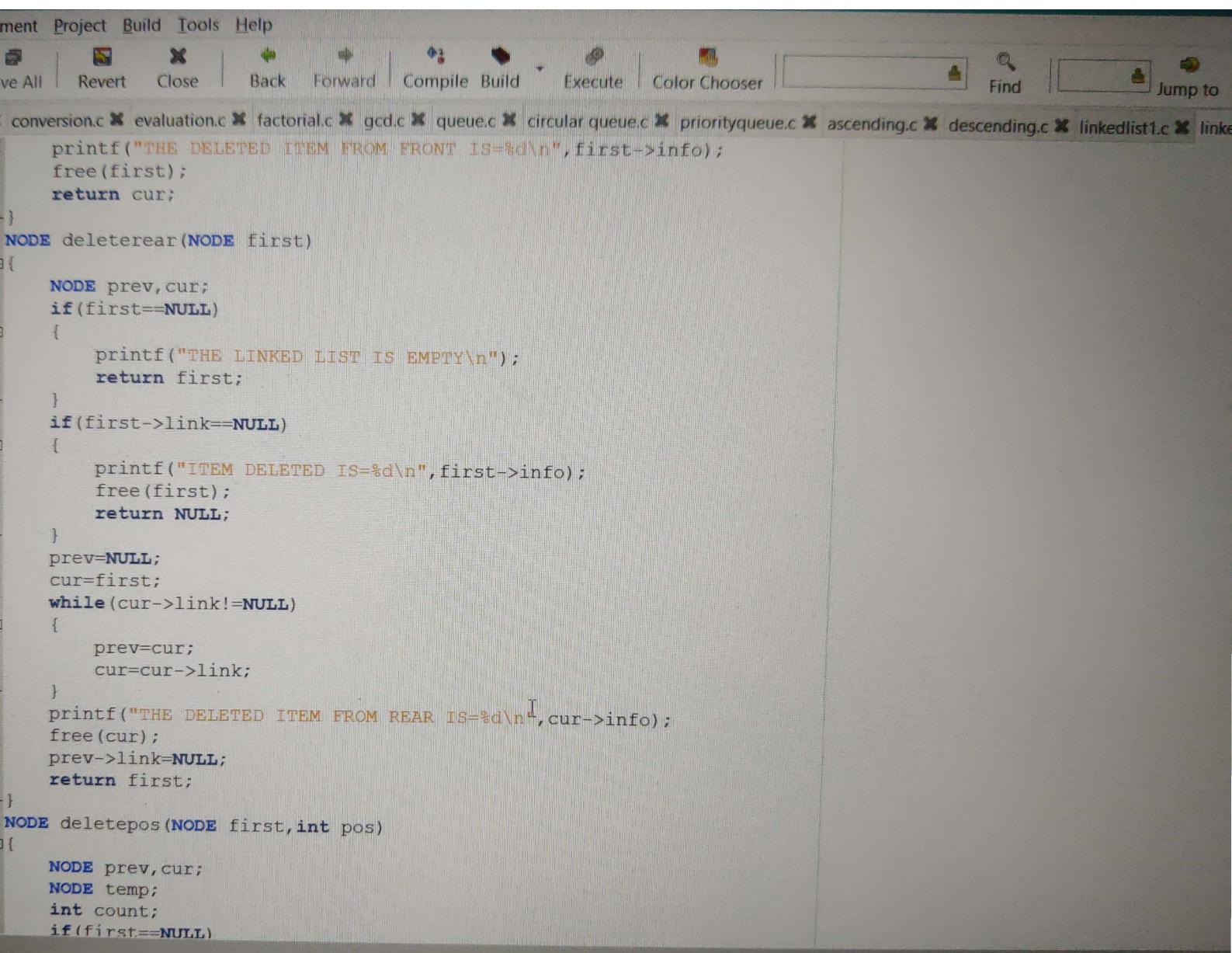
Save Save All Revert Close Back Forward Compile Build Execute Color Chooser Find Jump to Quit

stack.c conversion.c evaluation.c factorial.c gcd.c queue.c circular queue.c priorityqueue.c ascending.c descending.c linkedlist1.c linkedlist2.c

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

```
71 }
72 if(pos==1)
73 {
74     temp->link=first;
75     first=temp;
76     return temp;
77 }
78 count=1;
79 prev=NULL;
80 cur=first;
81 while(cur!=NULL&&count!=pos)
82 {
83     prev=cur;
84     cur=cur->link;
85     count++;
86 }
87 if(count==pos)
88 {
89     prev->link=temp;
90     temp->link=cur;
91     return first;
92 }
93 printf("invalid position\n");
94 return first;
95 }
96 }

97 NODE deletefront(NODE first)
98 {
99     NODE cur;
100    if(first==NULL)
101    {
102        printf("THE LINKED LIST IS EMPTY\n");
103        return first;
104    }
105    cur=first;
```



File Edit Search View Document Project Build Tools Help

New Open Save Save All Revert Close Back Forward Compile Build Execute Color Chooser Find Jump to Quit

Symbols Documents stack.c conversion.c evaluation.c factorial.c gcd.c queue.c circular queue.c priorityqueue.c ascending.c descending.c linkedlist1.c linkedlist2.c

Functions

- deletefront [98]
- deletapos [138]
- deleterear [112]
- display [176]
- getnode [10]
- insertfront [23]
- insertpos [56]
- insertrear [37]
- main [191]

Structs

- node [5]
 - info [6]
 - link [7]

Typedefs / Enums

- NODE [9]

```
138 NODE deletapos(NODE first,int pos)
139 {
140     NODE prev,cur;
141     NODE temp;
142     int count;
143     if(first==NULL)
144     {
145         printf("THE LINKED LIST IS EMPTY\n");
146         return NULL;
147     }
148     if(pos==1)
149     {
150         temp=first;
151         printf("THE DELETED ITEM FROM FRONT IS=%d\n",temp->info);
152         free(temp);
153         first=first->link;
154         return first;
155     }
156     count=1;
157     prev=NULL;
158     cur=first;
159     while(cur!=NULL && count!=pos)
160     {
161         prev=cur;
162         cur=cur->link;
163         count++;
164     }
165     if(count==pos)
166     {
167         printf("THE DELETED ITEM AT POSITION %d=%d\n",pos,cur->info);
168         prev->link=cur->link;
169         free(cur);
170         return first;
171     }
172     printf("INVALID CHOICE\n");
173     return first;
```

Status gcc -Wall -o "linkedlist2" "linkedlist2.c" (in directory: C:\Users\Nithin')
Compilation finished successfully.

Compiler

Messages

Scribble

line: 8 / 255 col: 2 sel: 0 INS TAB mode: CR/LF encoding: UTF-8 status: C

File Edit Search View Document Project Build Tools Help

New Open Save Save All Revert Close Back Forward Compile Build Execute Color Chooser Find Jump to Quit

Symbols Documents stack.c conversion.c evaluation.c factorial.c gcd.c queue.c circular queue.c priorityqueue.c ascending.c descending.c linkedlist1.c linkedlist2.c

Functions

- deletefront [98]
- deletetop [138]
- deleterear [112]
- display [176]
- getnode [10]
- insertfront [23]
- insertpos [56]
- insertrear [37]
- main [191]

Structs

- node [5]
 - info [6]
 - link [7]

Typedefs / Enums

- NODE [9]

```
171     }
172     printf("INVALID CHOICE\n");
173     return first;
174 }

175 void display(NODE first)
176 {
177     NODE temp;
178     if(first==NULL){
179         printf("THE LIST IS EMPTY\n");
180     }
181     printf("THE ELEMENTS ARE=");
182     for(temp=first;temp!=NULL,temp=temp->link)
183     {
184         printf("%d\t",temp->info);
185     }
186     printf("\n");
187 }
188 }

189 int main()
190 {
191     int c,item,pos;
192     NODE first=NULL;
193     for(;;)
194     {
195         printf("1-INSERTFRONT \n 2-INSERTREAR \n 3-INSERT AT GIVEN POSITION \n 4-DELETEFRONT \n 5-DELETEREAR \n 6-");
196         printf("ENTER THE CHOICE\n");
197         scanf("%d",&c);
198         switch(c)
199         {
200             case 1:
201                 printf("ENTER THE ELEMENT TO BE INSERTED FRONT\n");
202                 scanf("%d",&item);
203                 first=insertfront(first,item);
204                 break;
205             case 2:
206                 printf("ENTER THE ELEMENT TO BE INSERTED REAR\n");
207                 scanf("%d",&item);
208                 insertrear(first,item);
209                 break;
210             case 3:
211                 printf("ENTER THE ELEMENT AND POSITION\n");
212                 scanf("%d %d",&item,&pos);
213                 insertat(first,item,pos);
214                 break;
215             case 4:
216                 first=deletefront(first);
217                 break;
218             case 5:
219                 first=deleterear(first);
220                 break;
221             case 6:
222                 exit(0);
223         }
224     }
225 }
```

Status gcc -Wall -o "linkedlist2" "linkedlist2.c" (in directory: C:\Users\Nithin')
Compiler Compilation finished successfully.

Messages

Scribble

line: 8 / 255 col: 2 sel: 0 INC TAB

File Edit Search View Document Project Build Tools Help

New Open Save Save All Revert Close Back Forward Compile Build Execute Color Chooser Find Jump to Quit

Symbols Documents stack.c conversion.c evaluation.c factorial.c gcd.c queue.c circularqueue.c priorityqueue.c ascending.c descending.c linkedlist1.c linkedlist2.c

Functions

- deletefront [98]
- deletepos [138]
- deletrear [112]
- display [176]
- getnode [10]
- insertfront [23]
- insertpos [56]
- insertrear [37]
- main [191]

Structs

- node [5]
 - info [6]
 - link [7]

Typedefs / Enums

- NODE [9]

```
195 for(;;)
196 {
197     printf("1-INSERTFRONT \n 2-INSERTREAR \n 3-INSERT AT GIVEN POSITION \n 4-DELETEFRONT \n 5-DELETETEAR \n 6-DELETEPOS \n 7-");
198     printf("ENTER THE CHOICE\n");
199     scanf("%d", &c);
200     switch(c)
201     {
202         case 1:
203             printf("ENTER THE ELEMENT TO BE INSERTED FRONT\n");
204             scanf("%d", &item);
205             first=insertfront(first,item);
206             break;
207         case 2:
208             printf("ENTER THE ELEMENT TO BE INSERTED AT THE END\n");
209             scanf("%d", &item);
210             first=insertrear(first,item);
211             break;
212         case 3:
213             printf("ENTER THE ELEMENT AND THE POS AT WHICH IT SHOULD BE INSERTED\n");
214             scanf("%d", &item);
215             scanf("%d", &pos);
216             first=insertpos(item,pos,first);
217             break;
218         case 4:
219             first=deletefront(first);
220             break;
221         case 5:
222             first=deletrear(first);
223             break;
224         case 6:
225             printf("ENTER THE POS AT WHICH ELEMENT SHOULD BE DELETED\n");
226             scanf("%d", &pos);
227             first=deletepos(first,pos);
228             break;
229         case 7:
230             display(first);
```

```
conversion.c ✘ evaluation.c ✘ factorial.c ✘ gcd.c ✘ queue.c ✘ circular queue.c ✘ priorityqueue.c ✘ ascending.c ✘ descending.c ✘ linkedlist1.c ✘ linkedlist2.c ✘
break;
case 5:
first=deletelast(first);
break;
case 6:
printf("ENTER THE POS AT WHICH ELEMENT SHOULD BE DELETED\n");
scanf("%d",&pos);
first=deletepos(first,pos);
break;
case 7:
display(first);
break;
case 8:
exit(0);
default:
printf("INVALID CHOICE\n");
}
}
```

I

```
1-INSERTFRONT  
2-INSERTREAR  
3-INSERT AT GIVEN POSITION  
4-DELETEFRONT  
5-DELETEREAR  
6-DELETEPOS  
7-DISPLAY  
8-EXIT  
ENTER THE CHOICE  
1  
ENTER THE ELEMENT TO BE INSERTED FRONT  
23  
&& count  
1-INSERTFRONT  
2-INSERTREAR  
3-INSERT AT GIVEN POSITION  
4-DELETEFRONT  
5-DELETEREAR  
6-DELETEPOS  
7-DISPLAY  
8-EXIT  
ENTER THE CHOICE  
1  
ENTER THE ELEMENT TO BE INSERTED FRONT  
45  
1-INSERTFRONT  
2-INSERTREAR  
3-INSERT AT GIVEN POSITION  
4-DELETEFRONT  
5-DELETEREAR  
6-DELETEPOS  
VALID CHOICE  
st;  
+2.c" (in directory: C:\Users\Nithin')
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
1-INSERTFRONT  
2-INSERTREAR  
3-INSERT AT GIVEN POSITION  
4-DELETEFRONT  
5-DELETEREAR  
6-DELETEPOS  
7-DISPLAY  
8-EXIT
```

ENTER THE CHOICE

1
ENTER THE ELEMENT TO BE INSERTED FRONT
43

```
1-INSERTFRONT  
2-INSERTREAR  
3-INSERT AT GIVEN POSITION  
4-DELETEFRONT  
5-DELETEREAR  
6-DELETEPOS  
7-DISPLAY  
8-EXIT
```

ENTER THE CHOICE

7
THE ELEMENTS ARE=43 45 23

```
1-INSERTFRONT  
2-INSERTREAR  
3-INSERT AT GIVEN POSITION  
4-DELETEFRONT  
5-DELETEREAR  
6-DELETEPOS  
7-DISPLAY
```

```
8-EXIT
ENTER THE CHOICE
4
THE DELETED ITEM FROM FRONT IS=43
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DELETEFRONT
5-DELETETEAR
6-DELETEPOS
7-DISPLAY
8-EXIT
ENTER THE CHOICE
5
THE DELETED ITEM FROM REAR IS=23
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DELETEFRONT
5-DELETETEAR
6-DELETEPOS
7-DISPLAY
8-EXIT
ENTER THE CHOICE
6
ENTER THE POS AT WHICH ELEMENT SHOULD BE DELETED
1
THE DELETED ITEM FROM FRONT IS=45
1-INSERTFRONT
2-INSERTREAR
```

```
C:\WINDOWS\SYSTEM32\cmd.exe
7-DISPLAY
8-EXIT
ENTER THE CHOICE
6
ENTER THE POS AT WHICH ELEMENT SHOULD BE DELETED
1
THE DELETED ITEM FROM FRONT IS=45
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DELETEFRONT
5-DELETETEAR
6-DELETEPOS
7-DISPLAY
8-EXIT
ENTER THE CHOICE
6
ENTER THE POS AT WHICH ELEMENT SHOULD BE DELETED
1
THE LINKED LIST IS EMPTY
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DELETEFRONT
5-DELETETEAR
6-DELETEPOS
7-DISPLAY
8-EXIT
ENTER THE CHOICE
8
in directory: C:\Users\Nithin')
```