

```
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("mem full\n");
16         exit(0);
17     }
18     return x;
19 }
20 void freenode(NODE x)
21 {
22     free(x);
23 }
24 NODE insert_near(NODE first,int item)
25 {
26     NODE temp,cur;
27     temp=getnode();
28     temp->info=item;
29     temp->link=first;
```

```
temp->link=NULL;
if(first==NULL)
    return temp;
cur=first;
while(cur->link!=NULL)
    cur=cur->link;
cur->link=temp;
return first;
}
NODE delete_front(NODE first)
{
NODE temp;
if(first==NULL)
{
printf("list is empty cannot delete\n");
return first;
}
temp=first;
temp=temp->link;
printf("item deleted at front-end is=%d\n",first->info);
free(first);
return temp;
}
void display(NODE first)
{
NODE temp;
if(first==NULL)
printf("list empty cannot display items\n");
for(temp=first;temp!=NULL;temp=temp->link)
{
```

main.c

```
59     printf("%d \n",temp->info);
60 }
61 }
62 NODE insert_front(NODE first,int item)
63 {
64     NODE temp;
65     temp=getnode();
66     temp->info=item;
67     temp->link=NULL;
68     if(first==NULL)
69     return temp;
70     temp->link=first;
71     first=temp;
72     return first;
73 }
74 NODE delete_front_s(NODE first)
75 {
76     NODE temp;
77     if(first==NULL)
78     {
79         printf("stack is empty cannot delete\n");
80         return first;
81     }
82     temp=first;
83     temp=temp->link;
84     printf("item deleted at front-end is=%d\n",first->info);
85     free(first);
86     return temp;
87 }
```



main.c

```
88 void display_s(NODE first)
89 {
90     NODE temp;
91     if(first==NULL)
92         printf("stack empty cannot display items\n");
93     for(temp=first;temp!=NULL;temp=temp->link)
94     {
95         printf("%d\n",temp->info);
96     }
97 }
98 int main()
99 {
100     int item,choice,pos;
101     NODE first=NULL;
102     system("cls");
103     for(;;)
104     {
105         printf("\n Queue operations :\n 1:Insert_rear\n"
106               "2:Delete_front\n 3:Display_list(Queue)\n\n Stack\n"
107               "operations \n 4:Insert_front\n 5: Delete_front \n"
108               "6:Display_list(Stack)\n 7:Exit \n\n");
109         printf("enter the choice \n");
110         scanf("%d",&choice);
111         switch(choice)
112         {
113             case 1:printf("enter the item at rear-end\n");
114             scanf("%d",&item);
115             first=insert_rear(first,item);
116             break;
```



main.c

```
113     break;
114     case 2:first=delete_front(first);
115     break;
116     case 3:display(first);
117     break;
118     case 4:printf("enter the item at front-end\n");
119     scanf("%d",&item);
120     first=insert_front(first,item);
121     break;
122     case 5:first=delete_front_s(first);
123     break;
124     case 6:display_s(first);
125     break;
126     default:exit(0);
127     break;
128 }
129 }
130 return 0;
131 }
```

Queue operations :

- 1:Insert_rear
- 2:Delete_front
- 3:Display_list(Queue)

Stack operations

- 4:Insert_front
- 5: Delete_front
- 6:Display_list(Stack)
- 7:Exit

enter the choice

1

enter the item at rear-end

23

Queue operations :

- 1:Insert_rear
- 2:Delete_front
- 3:Display_list(Queue)

Stack operations

- 4:Insert_front
- 5: Delete_front
- 6:Display_list(Stack)
- 7:Exit

enter the choice

1

enter the item at rear-end

67

```
2:Delete_front  
3:Display_list(Queue)
```

Stack operations

```
4:Insert_front  
5: Delete_front  
6:Display_list(Stack)  
7:Exit
```

enter the choice

2

item deleted at front-end is=23

Queue operations :

```
1:Insert_rear  
2:Delete_front  
3:Display_list(Queue)
```

Stack operations

```
4:Insert_front  
5: Delete_front  
6:Display_list(Stack)  
7:Exit
```

enter the choice

3

67

Queue operations :

```
1:Insert_rear  
2:Delete_front  
3:Display_list(Queue)
```

4:Insert_front
5: Delete_front
6:Display_list(Stack)
7:Exit

enter the choice

4

enter the item at front-end

58

Queue operations :

1:Insert_rear
2:Delete_front
3:Display_list(Queue)

Stack operations

4:Insert_front
5: Delete_front
6:Display_list(Stack)
7:Exit

enter the choice

4

enter the item at front-end

89

Queue operations :

1:Insert_rear
2:Delete_front
3:Display_list(Queue)

Stack operations

4:Insert_front

enter the choice

5
item deleted at front-end is=89

Queue operations :

1:Insert_rear
2:Delete_front
3:Display_list(Queue)

Stack operations

4:Insert_front
5: Delete_front
6:Display_list(Stack)
7:Exit

enter the choice

6
58
67

Queue operations :

1:Insert_rear
2:Delete_front
3:Display_list(Queue)

Stack operations

4:Insert_front
5: Delete_front
6:Display_list(Stack)
7:Exit

enter the choice

19:58
02-01-2021