File Edit Search Run Compile Debug Project Options Window Help

QUEUELIN.C

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
int info;
struct node *link;
}
*f=NULL,*r=NULL;
int main()
{
int c;
clrscr();
while(1)
{
printf("\n Main menu");
printf("1.insert\n");
printf("2.Delete\n");
printf("3.Display\n");
printf("4.Quit\n");
printf("enter the choice  :  ");
```

1:28

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

File Edit Search Run Compile Debug Project Options Window Help

QUEUELIN.C

```
printf("enter the choice  :  ");
scanf("%d",&c);
switch(c)
{
case 1:
push();
break;
case 2:
pop();
break;
case 3:
display();
break;
case 4:
exit(1);
break;
default:
printf("wrong choice\n");
}
}
}
```


41:28

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```
≡ File Edit Search Run Compile Debug Project Options Window Help
[■]═══ QUEUELIN.C ═══2=[↑]
}
push()
{
struct node* t;
int item;
t=(struct node*)malloc(sizeof(struct node));
printf("Input the element to be insertedon the top :  ");
scanf("%d",&item);
t->info=item;
t->link=NULL;
if(f==NULL)
f=t;
else
r->link=t;
r=t;
return;
}
pop()
{
struct node *t;
if(f==NULL)
═══ 61:28 ═══
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

```
File  Edit  Search  Run  Compile  Debug  Project  Options  Window  Help
[■]                    QUEUELIN.C                              2=[↑]
if(f ==NULL)
printf("queue empty");
else
t=f;
printf("delete the %d element ",t->info);
f=f->link;
free(t);
return;
}
display()
{
struct node *ptr;
ptr=f;
if(f ==NULL)
printf("queue is empty");
else
printf("queue elements  :  \n");
while(ptr!=NULL)
{
printf("%d\n",ptr->info);
ptr=ptr->link;
        81:28
F1 Help  F2 Save  F3 Open  Alt-F9 Compile  F9 Make  F10 Menu
```

```
printf("delete the %d element ",t->info);
f=f->link;
free(t);
return;
}
display()
{
struct node *ptr;
ptr=f;
if(f==NULL)
printf("queue is empty");
else
printf("queue elements  :  \n");
while(ptr!=NULL)
{
printf("%d\n",ptr->info);
ptr=ptr->link;
}
printf("\n");
return;
}
```

```
 Main menu1.insert
2.Delete
3.Display
4.Quit
enter the choice  :  1
Input the element to be insertedon the top :  5

 Main menu1.insert
2.Delete
3.Display
4.Quit
enter the choice  :  3
queue elements  :
5


 Main menu1.insert
2.Delete
3.Display
4.Quit
enter the choice  :  _
```

PROGRAM: 7

1) Write a program of queue ins using linked list

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
Struct node
{
int info;
Struct node *link;
};
* F = NULL, *r = NULL;
int main()
{
int c;
clrscr();
while (1)
{
printf("\n Main menu");
printf ("1. insert\n");
printf ("2. delete\n");
printf ("3. Display\n");
printf ("4. Quit.");
printf (" Enter the choice :");
scanf ("%d", &c);
Scoitch (c)
{
case 1:
        push();
        break;
case 2:
```

```
pop();
break;
case 3:
display();
break;
case 4:
exit(1);
break;
default;
printf("wrong choice\n");
}
}
}
push()
{
struct node *t;
int item;
t=(struct node *)malloc(sizeof(struct node));
printf("input the element to be inserted :");
scanf("%d",&item);
t->info=item;
t->link=NULL;
IF(f==NULL)
f=t;
else
r->link=t;
r=t;
return;
}
pop()
{
struct node *t;
```

```
IF (F == NULL)
printf ("queue empty");
else
t = f;
printf (" delete the %d element ", t -> info);
f = f -> link;
free (t);
return;
 }
display()
}
Struct node * ptr;
ptr = f;
 IF (f == NULL)
printf (" queue is empty ");
 else
printf (" que elements. : \n");
while (ptr != NULL)
 {
printf ("%d \n", ptr -> info);
 ptr = ptr -> link;
  }
printf ("\n");
 return;
  }
```

OUTPUT

Main menu:
1. Insert
2. Delete
3. Display
4. Quit
Enter the choice : 1
Input the element to be insert the top: 9

Main menu
1. Insert
2. Delete
3. Display
4. Quit
Enter the choice : 3
Queue elements : 5