

Program no: 8

Program to implement queue using linked list

Program

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

struct node
{
    int data info;
    struct node *link;
}

*F = NULL, *R = NULL;

int main()
{
    int c;
    clrscr();
    while (1)
    {
        printf("\n MAIN MENU");
        printf("\n 1. Insert");
        printf("\n 2. delete");
        printf("\n 3. display");
        printf("\n 4. Quit");
    }
}
```

```

printf("\n enter the choice");
scanf("%d", &c);
switch(c)
{
case 1: push(c);
        break;
case 2: pop(c);
        break;
case 3: display(c);
        break;
case 4: exit(c);
        break;
default: printf("wrong");
}
}
}
push(c)
{
struct node * t;
int item;
t = (struct node *) malloc (sizeof (struct node));
printf("\n insert the element");
scanf("%d", &item);

```

```

t -> info = item;
t -> link = NULL;
if (t == NULL)
    F = t;
else
    r -> link = t;
    r = t;
return;
}

pop()
{
    struct node *t;
    if (t == NULL)
        printf("Queue empty");
    else
    {
        t = t;
        printf("Delete the %d element", t -> info);
        t = t -> link;
        free(t);
    }
    return;
}

```



```

display()
{
    struct node *ptr;
    ptr = f;
    if (f == NULL)
        printf("In queue is empty");
    else
    {
        printf("queue element are: \n");
        while (ptr != NULL)
        {
            printf("%d\n", ptr->data);
            ptr = ptr->link;
        }
        printf("%d\n", ptr->data);
        ptr = ptr->link;
    }
    printf("\n");
}
return;
}

```

output

MAIN MENU

1. insert
2. delete
3. display
4. quit

enter the choice 1

insert the element 5

MAIN MENU

1. insert
2. delete
3. display
4. quit

enter the choice 3

queue element are:

5

MAIN MENU

1. insert
2. delete
3. display
4. quit

enter the choice

```
#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("\nMAIN MENU");
```

```
printf("\n1.Insert");
```

```
printf("\n2.delete");
```

```
printf("\n3.display");
```

```
printf("\n4.Quit");
```

```
printf("\n enter the choice");
```



```
[■]===== \TURBOC3\SARANYA\QUEUE\IN.C =====2=[↑]
printf("\n 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");
}
}
}
push()
{
struct node* t;
int item;
t=(struct node*)malloc(sizeof(struct node));
26:68
```

```
printf("\n insert the element");
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("\nqueue empty");
else
{
t=f;
printf("\ndelete the %d element ",t->info);
f=f->link;
```



```
[■] \TURBOC3\SARANYA\QUEUE\IN.C 2=[↑]
free(t);
}
return;
}
display()
{
struct node *ptr;
ptr=f;
if(f==NULL)

printf("\n queue is empty");
else
{
printf("queue element are:\n");
while(ptr!=NULL)
{
printf("%d\n",ptr->info);
ptr=ptr->link;
}
printf("\n");
}
}
68:68
```

```
[■]===== \TURBOC3\SARANYA\QUEUE\IN.C =====2=[↑]
}
display()
{
struct node *ptr;
ptr=f;
if(f==NULL)

printf("\n queue is empty");
else
{
printf("queue element are:\n");
while(ptr!=NULL)
{
printf("%d\n",ptr->info);
ptr=ptr->link;
}
printf("\n");
}
return;
}
```

MAIN MENU

- 1.Insert
- 2.delete
- 3.display
- 4.Quit

enter the choice 1

insert the element 5

MAIN MENU

- 1.Insert
- 2.delete
- 3.display
- 4.Quit

enter the choice 3

queue element are:

5

MAIN MENU

- 1.Insert
- 2.delete
- 3.display
- 4.Quit

enter the choice