Prog-8

Write a Program to implement Queue 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("\n 1. insert");
  printf("\n 2. delete");
  printf("\n 3. display");
  printf("\n 4. Quit");
  printf("\n Enter the choice")
  scanf("\n %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));
printf ("In insert the element");
scanf ("%d", &item);
t -> info = item;
t -> link = NULL;
```

```
if (f == NULL)
   f=ti
   else
r->link=ti
   r=ti
   return;
}
POP()
{ Struct node *t;
   if (f==NULL)
   printf ("\n queue is empty");
   else
   { t=f;
Printf("\n delete the 1st element ", t-> info);
   f= f->link;
      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");
}
return;
  }
```

OUTPUT

MAIN MENU

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

Enter the choice 1

insert the element : 3

MAIN MENU

1. insert
2. delede
3. display
4. exit

Enter the choice : 3

Queue element are :

3

```
#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");
```

```
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));
```

```
t=(struct node*)malloc(sizeof(struct node));
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);
```

```
printf("\ndelete the %d element ",t->info);
f=f->link;
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("\ndelete the %d element ",t->info);
f=f->link;
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;
}
```

```
}
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 3

MAIN MENU
1.Insert
2.delete
3.display
4.Quit
 enter the choice 3
queue element are:
3


MAIN MENU
1.Insert
2.delete
3.display
4.Quit
 enter the choice
```