circular Queue program

```
#include<stdio.h>
#include<conio.h>
#define QUE_SIZE 5
int item, front=0, rear=-1, q[QUE_SIZE], count=0;
void insertrear()
{ if(count==QUE_SIZE)
{ printf("queue overflow\n");
 return;
}
 rear=(rear+1)%QUE_SIZE;
 q[rear]=item;
 count++;
}
 int deletefront()
{ if(count=0) return -1;
item=q[front];
front=(front+1)%QUE_SIZE;
count=count-1;
return item;
}
void display()
{
 int i;
 if(count==0)
{ printf("queue is empty\n");
 return;
```

```
printf ("Contents of queue \n");
 for (i=1; i<=count; i++)
{ printf("%d \n", q[f]);
  front = (front+1) % QUE_SIZE;
}
}
void main ()
{ int choice;
   for(;;)
{ printf("\n1: insertrear \n2: deletefront \n3: display
                                \n4: exit \n");
printf ("Enter the choice \n");
scanf ("%d", &choice);
Switch (choice)
{ case 1 : printf ("Enter the item to be inserted \n");
           Scanf ("%d", &item);
           insertrear();
           break;
case 2:  item = deletefront()
         if (item == -1)
         printf ("Q is empty");
         else
         priatf ("item ddeted = %d\n", "item);
         break;
case 3:  display@();  break;
default : exit (0);
}}
```

main.c

```c
#include<stdio.h>
#include<stdlib.h>
#define QUE_SIZE 3
int item,front=0,rear=-1,q[QUE_SIZE],count=0;
void insertrear()

{
if(count==QUE_SIZE)
{
printf("queue overflow\n");
return;
}
rear=(rear+1)%QUE_SIZE;
q[rear]=item;
count++;
}

int deletefront()

{
if(count==0) return -1;
item=q[front];
front=(front+1)%QUE_SIZE;
count=count-1;
return item;
}

void displayQ()

{
int i,f;
if(count==0)
{
printf("queue is empty\n");
return;
}
f=front;
printf("Contents of queue \n");
for(i=1;i<=count;i++)
{
printf("%d\n",q[f]);
f=(f+1)%QUE_SIZE;
}
}

void main()
{
 int choice;

 for(;;)
 {
 printf("\n1:insertrear\n2:deletefront\n3:display\n4:exit\n");
 printf("enter the choice\n");
 scanf("%d",&choice);
```

main.c

```
{
 int choice;

 for(;;)
 {
 printf("\n1:insertrear\n2:deletefront\n3:display\n4:exit\n");
 printf("enter the choice\n");
 scanf("%d",&choice);

 switch(choice)
 {
 case 1:printf("enter the item to be inserted\n");
    scanf("%d",&item);
    insertrear();
    break;
 case 2:item=deletefront();
    if(item==-1)
    printf("queue is empty\n");
    else
    printf("item deleted =%d\n",item);
    break;
 case 3:displayQ();
    break;
 default:exit(0);
 }
 }
 getch();
 }
```

input

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
7

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
8

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
```

```
enter the item to be inserted
9

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
7
8
9

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
4
queue overflow

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =7

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =8

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =9

1:insertrear
2:deletefront
3:display
4:exit
```

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =9

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
queue is empty

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
5

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
4


...Program finished with exit code 0
Press ENTER to exit console.
```