

19/10/20

POOJA K

1BM19CS111

// Circular queue program.

#include <stdio.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;

```

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;
}

int main()
{
    int choice;
    for (;;)
    {
        printf("\n 1: insert rear \n 2: delete front\n"
               "\n 3: display\n 4: 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");
        }
    }
}

```

POOJA.K
18M19CS111

else

printf("item deleted <id> \n", item);

break;

case 3 : displayQ();

break;

default : exit(0);

}

}

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define que_size 3
4 int item,front=0,rear=-1,q[que_size],count=0;
5 void insertrear()
6 {
7     if(count==que_size)
8     {
9         printf("queue overflow");
10    return;
11 }
12 rear=(rear+1)%que_size;
13 q[rear]=item;      I
14 count++;
15 }
16 int deletefront()
17 {
18     if(count==0) return -1;
19     item = q[front];
20     front=(front+1)%que_size;
21     count=count-1;
22     return item;
23 }
24 void displayq()
25 {
```



main.c

```
44     void display()
45 {
46     int i,f;
47     if(count==0)
48     {
49         printf("queue is empty");
50         return;
51     }
52     f=front;
53     printf("contents of queue \n");
54     for(i=0;i<count;i++)
55     {
56         printf("%d\n",q[f]);
57         f=(f+1)%que_size;
58     }
59 }
60 int main()    T
61 {
62     int choice;
63     for(;;)
64     {
65         printf("\n1.Insert rear \n2.Delete front
66 \n3.Display \n4.exit \n ");
67         printf("Enter the choice : ");
68         scanf("%d",&choice);
```



```
13.Display \n4.exit \n ");
46 printf("Enter the choice : ");
47 scanf("%d",&choice);
48 switch(choice)
49 {
50     case 1:printf("Enter the item to be inserted :");
51         scanf("%d",&item);
52         insertrear();
53         break;
54     case 2:item=deletefront();
55         if(item==-1)
56             printf("queue is empty\n");
57         else
58             printf("item deleted is %d \n",item);
59         break;
60     case 3:displayq();
61         break;
62     default:exit(0);
63 }
64 }
65 return 0;
66 }
```

```
> clang-7 -pthread -lm -o main main.c  
> ./main
```

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 1

Enter the item to be inserted :10

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 1

Enter the item to be inserted :20

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 1

Enter the item to be inserted :30

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 3

contents of queue

```
Enter the choice : 3
contents of queue
10
20
30
```

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

```
Enter the choice : 2
item deleted is 10
```

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

```
Enter the choice : 1
Enter the item to be inserted :40
```

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

```
Enter the choice : 3
contents of queue
20
30
40
```

Enter the choice : 3
contents of queue
10
20
30

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2
item deleted is 10

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 1
Enter the item to be inserted :40

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 3
contents of queue
20
30
40

40

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 1

Enter the item to be inserted :50

queue overflow

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2

item deleted is 20

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2

item deleted is 30

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2

item deleted is 40

Enter the choice : 2

item deleted is 20

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2

item deleted is 30

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2

item deleted is 40

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 2

queue is empty

- 1.Insert rear
- 2.Delete front
- 3.Display
- 4.exit

Enter the choice : 4

> |