circularqueue - Notepad

File Edit Format View Help

```
#include<stdio.h>
#include<stdlib.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 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;
}
```

```
}
int main(){
int choice;
for(;;){
printf("\n1.insertrear\n2.deletefront\n3.display\n4.exit\n");
printf("Enter choice:\n");
scanf("%d",&choice);
switch(choice){
case 1:printf("Enter item to be inserted:\n");
scanf("%d",&item);
insertrear();
break;
case 2:item=deletefront();
if(item==-1)
printf("Queue empty!\n");
else
printf("Item deleted=%d\n",item);
break;
case 3:displayQ();
break;
default:exit(0);
return 0;
}
}
}
```

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

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
10

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
20

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
30

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
```

```
40

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
50

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
60
Queue Overflow!

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=10

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=20
```

```
1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
3
Contents of queue:
30
40
50

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=30

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=40

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=50

1.insertrear
2.deletefront
```

```
3.display
4.exit
Enter choice:
2
Queue empty!

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
4
```

dequeue - Notepad

File Edit Format View Help

```
#include<stdio.h>
#include<stdlib.h>
#define qsize 5
int f=0,r=-1,ch;
int item,q[10];

int isfull()
  {
   return(r==qsize-1)?1:0;
  }
int isempty()
  {
   return(f>r)?1:0;
  }
void insert_rear()
  {
   if(isfull())
                {
                 printf("queue overflow\n");
                 return;
                }
   r=r+1;
   q[r]=item;
  }
void delete_front()
  {
   if(isempty())
                {
                 printf("queue empty\n");
                 return;
                }
   printf("item deleted is %d\n",q[(f)++]);
   if(f>r)
```

dequeue - Notepad

File Edit Format View Help

```
                {
                 f=0;
                 r=-1;
                }
 }
void insert_front()
 {
 if(f!=0)
                {
                 f=f-1;
                 q[f]=item;
                 return;
                }
               else if((f==0)&&(r==-1))
                {
                 q[++(r)]=item;
                 return;
                }
               else
                 printf("insertion not possible\n");
 }
void delete_rear()
 {
 if(isempty())
                {
                 printf("queue is empty\n");
                 return;
                }
 printf("item deleted is %d\n",q[(r)--]);
 if(f>r)
                {
                 f=0;
                 r=-1;
```

dequeue - Notepad

File Edit Format View Help

```
            }
  }
void display()
  {
   int i;
   if(isempty())
            {
             printf("queue empty\n");
             return;
            }
    for(i=f;i<=r;i++)
            printf("%d\n",q[i]);
  }
int main()
 {

  for(;;)
   {
            printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
            printf("enter choice\n");
            scanf("%d",&ch);
            switch(ch)
             {
              case 1:printf("enter the item\n");
                                scanf("%d",&item);
                                insert_rear();
                                break;
              case 2:printf("enter the item\n");
                                scanf("%d",&item);
                                insert_front();
                                break;
              case 3:delete_rear();
                                break;
              case 4:delete_front();
                                break;
              case 5:display();
                                break;
              default:exit(0);
             }
            }
  return 0;
  }
```

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

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
10

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
20

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
30

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
```

```
40

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
50

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
1
Enter item to be inserted:
60
Queue Overflow!

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=10

1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
2
Item deleted=20
```

```
1.insertrear
2.deletefront
3.display
4.exit
Enter choice:
3
Contents of queue:
30
40
50

1.insertrear
2.deletefront
> clang-7 -pthread -lm -o main main.c
> ./main
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear
2.insert_front
```

```
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
```

```
enter the item
60
queue overflow
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
10
insertion not possible
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
10
20
30
40
1.insert_rear
2.insert_front
```

```
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
20
30
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
❯ clang-7 -pthread -lm -o main main.c
❯ ./main
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```