28/10/2020

LP #4 WAP to implement ~~double ended~~ circular Queue ~~(dequeue)~~

```
#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++;
}
```

Scanned with CamScanner

```
int delfront()
{
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;
}
}
```

Scanned with CamScanner

```
int main()
 {
int choice;
printf ("\n1. insertrear \n2. deletefront \n3. display \n4. exit\n");
do
{
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 : break;
 }
 }
```

Scanned with CamScanner

```
while (choice != 4);
return 0;
}
```

Scanned with CamScanner