1BM19CS143
Sanjiahi K.
~~Sanjidha~~ k
Date: 19.10.20.

//Circular Queue program.

```
#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("\n 1. insertrear\n 2. deletefront \n 3. display \n
4. 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 is empty ! \n");
else
printf("Item deleted = %d \n", item);
break;
case 3: displayQ();
break;
default: exit(0);
return 0;
}
}
}
```