IBM19CS006

6/11/20 Lab 4 - Circular Queue

```
#include <stdio.h>
#include <stdlib.h>
int front=-1, rear=-1;
int queue[20];
int max;

void enque(int);
void deque();
void display();
int main()
{
  int choice, item;
  printf("Enter the size of the queue: ");
  scanf("%d", &max);

do{
  printf("\n1. Insert element");
  printf("\n2. Delete element");
  printf("\n3. Display queue");
  printf("\n4. Exit\n");
  printf("Enter your choice: \n");
  scanf("%d", &choice);

switch(choice){
case 1: if((front==0 && rear==max-1) ||
          (front=rear+1))
  { printf("Queue full\n");
  }
```

```
else {
    printf ("Enter element \n");
    scanf ("%d", &item);
    enque (item);
    break;

case 2: deque ();
        break;
case 3: display ();
        break;
case 4: exit (0);
default : printf ("\n Invalid choice \n");
}
} while (choice != 4);
return 0;
}

void enque (int ele)
{
    rear = (rear+1) % max;
    queue [rear] = ele;
    if (front == -1)
        front = 0;
    return;
}
```

1BM19CS006

```
void deque()
{
  int item;
  if ((front == -1) && (rear == -1))
  {
  printf ("Queue is empty \n");
   return;
  }
  else {
    item = queue[front];
  if (front == rear)
    {
      front = -1;  rear = -1;
    }
  else {
    front = (front+1) % max;
  }
}
printf ("Deleted element is : %d \n", item);
  return;
  }
void display()
  { int i;
  if ((front == -1) && (rear == -1))
    { printf ("Queue empty \n");
      return;
    }
```

IBM19CS006

```
else {
printf ("Queue contents : \n");
for (i = front; i != rear; i = (i+1) % max)
    {
    printf ("%d ", queue[i]);
    }
printf ("%d . ", queue[rear]);
return;
}
}
```