DAYANAND
1BM19CS043
6-11-2020

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 3

    int   front =-1
    int   rear =-1

int queue [MAX];

 void enqueue (int);
  int Deque ();
   void display ();
   int main (int argc, char **argv)
   {
     int option;
      int item;
   do{ printf("circular queue\n");
   printf(" circular queue\n");
   printf("\n1. Insert the queue(Enqueue)");
   printf("\n 2. delete from the queue (Dequeue)");
   printf("\n3. Display the content");
    printf("\n 4. Exit\n");
   scanf (" %d", &option);
    switch (option)
   {
     case 1 : printf ("enter the element \n");
              scanf ("%d", &item)
              Enque (item)
               break;
```

Scanned by TapScanner

```
case 2: item = Dequeue();
        if (item == -999)
          Printf("Queue is empty");
    else
      Printf("Removed element from the queue %d", item);
          break;
case 3: display();
          break;
case 4: exit(0);
}
} while (option != 4);
    return 0;
}
void enqueue(int ele)
{
  if ((front == 0 && rear == MAX-1) || (front == rear+1))
{
  Point("Queue is full\n"); return;
}
  else
{ rear = (rear+1) % MAX;
queue[rear] = ele;
if (front == -1)
  front = 0;
```

Scanned by TapScanner

```
int Deque()
{
   int item;
   if((front == -1) && (rear == -1))
   {
      return(-999);
   }
   else
   {item = queue[front];
    if (front == rear)
    {  front = -1;
       rear = -1;
    }
    else
       front=(front+1)%MAX;
    }
      return item;
   }
}
void display()
{  int i;
   if((front==-1)&&(rear==-1)||(front==rear))
   {  printf("queue is empty\n"); return;
   }
   else
{ printf("\n queue contents:\n");
  for (i=front; i<=rear; i++)
     printf("%d", queue[i]); }}
```

Scanned by TapScanner