LAB-04

Date : 08-01-24
Page No :

1. WAP to stimulate the working of the queue of integers using an array. Provide the following operations: Insert, delete, display. The program should print appropriate message for overflow and underflow condition.

```
#includ<stdio.h>
#define N 3

int queue[N], front=-1, rear=-1;

void enqueu(int n);
void dequeue();
void display();

void main()
{
    int choice, num;

    while(1){
        printf("Enter the operation:\n1.enqueue\n
                2.dequeue\n3.display\n4.-1to stop");
        scanf("%d", &choice);
    if(choice==-1){
        printf("operation completed.\n");
        break;
    }
    else{
        switch(choice)
        {
        case 1: printf("Enter the number:\n");
```

```
            scanf("%d", &num);
            enqueue(num);
            break;
        case 2: dequeue();
            break;
        case 3: display();
            break;
        default: printf("invalid input");
        }
    }
}
}

void enqueue(int n)
{
    if(rear == N-1)
    {
        printf("queue overflow\n");
    }
    else if(front == -1 && rear == -1)
    {
        front = rear = 0;
        queue[rear] = n;
        printf("succesfully enqueued\n");
    }
    else
    {
        rear ++;
        queue[rear] = n;
        printf("succesfully enqueued\n");
    }
}
```

```
void dequeUP()
{
  if(front==-1)
  {
    printf("queue underflow\n");
  }
  else if(front==rear)
  {
    printf("deleted element is :%d\n",queue[front]);
    front=rear=-1;
  }
  else
  {
    printf("deleted element is :%d\n",queue[front]);
    front++;
  }
}

void display()
{
  if(front==-1)
  {
    printf("queue is empty\n");
  }

  else
  {
    int i;

    printf("Elements are:\n");
```

```
        for(i=front; i<=rear; i++)
        {
            printf("%d\n", queue[i]);
        }
    }
}
```

//output

Enter operation
1. enque.
2. dequeue
3. display
4. -1 to stop
1
Enter the number:
7
succesfully enqueued
Enter the operation:
1. enqueue
2. dequeue
3. display
4. -1 to stop
8
succesfully enqueued
Enter operation
1. enqueue
2. dequeue
3. display
4. -1 to stop
3.
Elements are
7
8

C:\Users\bmsce\Desktop\22cs300\linque.exe

```
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
5
successfully enqueued
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
6
successful enqueued
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
7
successful enqueued
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
8
queue overflow
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
3
Elements are:
5
6
7
```

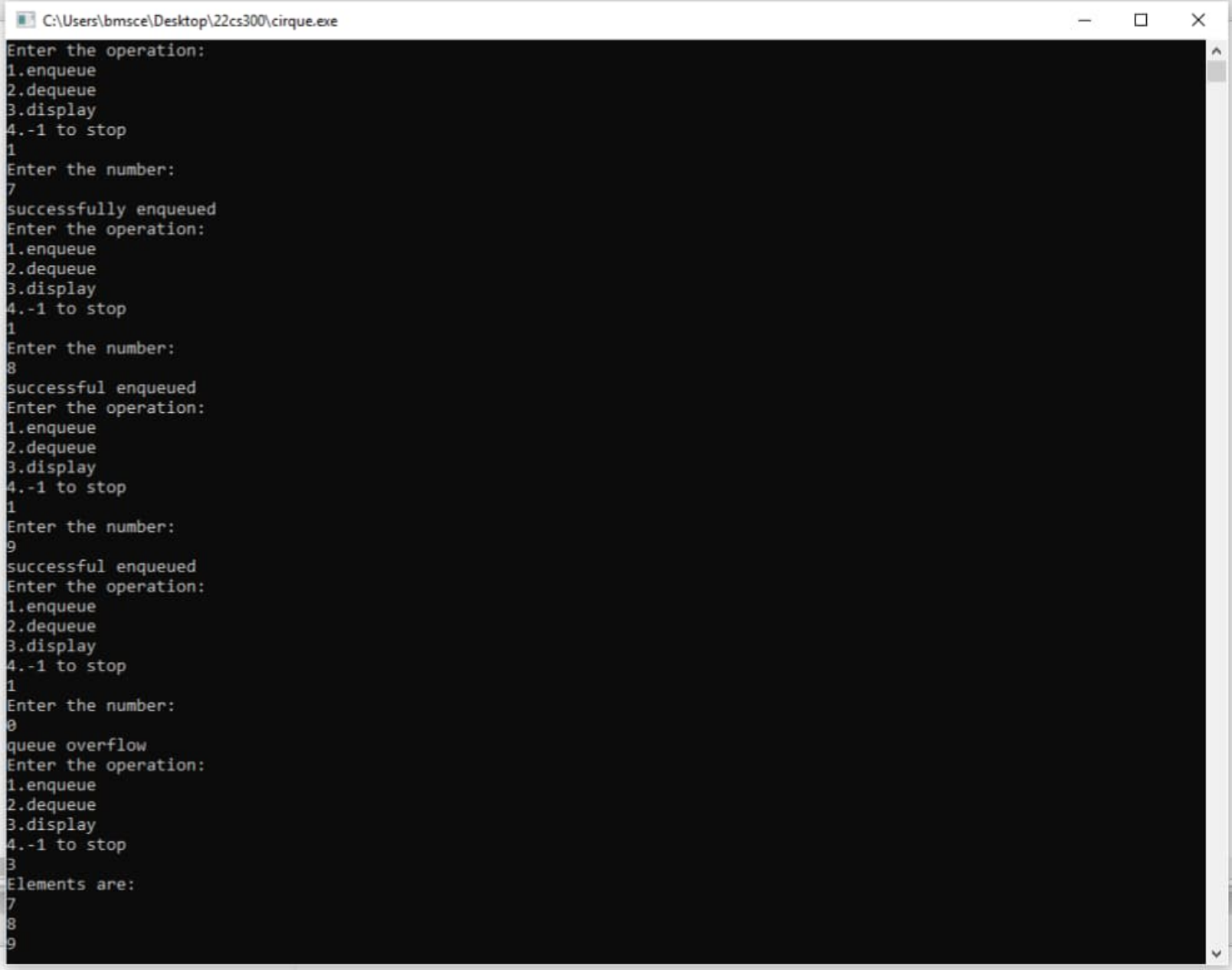
2. WAP to stimulate the working of the circular queue of integers using an array. Provide the following operations: Insert, delete, display. The program should print appropriate message for overflow and underflow condition.

```
#include <stdio.h>
#define N 3

int queue[N], front=-1, rear=-1;

void enque(int n);
void deque();
void display();

void main()
{
  int choice, num;

  while(1){
        printf("Enter the operation\n1. enqueue\n
              2.dequeue\n 3. display\n 4.-1 to stop\n");
        scanf("%d", &choice);
  if (choice==-1){
              printf("operation completed\n");
               break;
             }

  else{
     switch(choice)
     {
     case 1: printf("Enter the number\n");
```

```
            scanf("%d", &num);
            enqueue(num);
            break;
        case 2: dequeue();
            break;
        case 3: display();
            break;
        default: printf("invalid input\n");
    }
  }
 }
}


void enqueue(int n)
{
  if((rear+1)%N == front)
    {
      printf("queue overflow\n");
    }
  else if(front == -1 && rear == -1)
    {
      front = rear = 0;
      queue[rear] = n;
      printf("Successfully enqueued\n");
    }
  else
    {
      rear = (rear+1)%N;
      queue[rear] = n;
      printf("Successfully enqueued\n");
    }
}
```

```
void dequeue()
{
  if (front==-1)
    {
      printf("queue underflow\n");
    }
  else if (front==rear)
    {
      printf("deleted element is=%d\n",
                                queue[front]);
      front=rear=-1;
    }
  else
    {
      printf("deleted element is %d\n",
                                queue[front]);
      front=(front+1)%N;
    }
}

void display()
{
  int i; if (front==-1) { printf("queue empty\n");
else {printf("Elements are:\n");

  for (i=front; i!=rear; i=(i+1)%N)
    {
      printf("%d\n",queue[i]);
    }
}
}
```

//output
Enter the operation:
1. enqueue
2. dequeue
3. display
4. -1 to stop
1
Enter the number:
6
succesfully enqueued
Enter the operation:
1. enqueue
2. dequeue
3. display
4. -1 to stop
1
Enter the number:
7
succesfully enqueued
Enter the operation
1. enqueue
2. dequeue
3. display
4. -1 to stop
3
Elements are:
6
7


C:\Users\bmsce\Desktop\22cs300\cirque.exe
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
7
successfully enqueued
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
8
successful enqueued
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
9
successful enqueued
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
1
Enter the number:
0
queue overflow
Enter the operation:
1.enqueue
2.dequeue
3.display
4.-1 to stop
3
Elements are:
7
8
9