**Topic – Implementing Queue and Circular Queue Operations with the Help of Array**

* **Problem Statement**

Write C program to implement a  Queue  and a Circular Queue using array. Consider 1000 enqueue and dequeue operations at random ( Generate a random number 0 or 1 to identify enqueue or dequeue operation). Report the number of queuefull and queue empty conditions in each case after every 25 operations .

**Input example:**

The inputs are the random numbers which are equeued and random number 0 and 1 are chosen to identify enqueue and dequeue operations .

**Output example:**

\*\*\*\*\*The number of queuefull and queueempty conditions of a normal queue after successive 25 operations\*\*\*\*\*

The number of queuefull conditions are 0 after 25 operation

The number of queueempty conditions are 0 after 25 operation

The number of queuefull conditions are 1 after 50 operation

The number of queueempty conditions are 0 after 50 operation

The number of queuefull conditions are 2 after 75 operation

The number of queueempty conditions are 1 after 75 operation

The number of queuefull conditions are 2 after 100 operation

The number of queueempty conditions are 6 after 100 operation

The number of queuefull conditions are 2 after 125 operation

The number of queueempty conditions are 7 after 125 operation

The number of queuefull conditions are 3 after 150 operation

The number of queueempty conditions are 7 after 150 operation

The number of queuefull conditions are 3 after 175 operation

The number of queueempty conditions are 7 after 175 operation

The number of queuefull conditions are 3 after 200 operation

The number of queueempty conditions are 11 after 200 operation

The number of queuefull conditions are 3 after 225 operation

The number of queueempty conditions are 13 after 225 operation

The number of queuefull conditions are 3 after 250 operation

The number of queueempty conditions are 16 after 250 operation

The number of queuefull conditions are 4 after 275 operation

The number of queueempty conditions are 16 after 275 operation

The number of queuefull conditions are 4 after 300 operation

The number of queueempty conditions are 16 after 300 operation

The number of queuefull conditions are 4 after 325 operation

The number of queueempty conditions are 18 after 325 operation

The number of queuefull conditions are 4 after 350 operation

The number of queueempty conditions are 18 after 350 operation

The number of queuefull conditions are 4 after 375 operation

The number of queueempty conditions are 18 after 375 operation

The number of queuefull conditions are 4 after 400 operation

The number of queueempty conditions are 18 after 400 operation

The number of queuefull conditions are 4 after 425 operation

The number of queueempty conditions are 18 after 425 operation

The number of queuefull conditions are 4 after 450 operation

The number of queueempty conditions are 18 after 450 operation

The number of queuefull conditions are 4 after 475 operation

The number of queueempty conditions are 22 after 475 operation

The number of queuefull conditions are 4 after 500 operation

The number of queueempty conditions are 22 after 500 operation

The number of queuefull conditions are 5 after 525 operation

The number of queueempty conditions are 22 after 525 operation

The number of queuefull conditions are 5 after 550 operation

The number of queueempty conditions are 22 after 550 operation

The number of queuefull conditions are 5 after 575 operation

The number of queueempty conditions are 22 after 575 operation

The number of queuefull conditions are 5 after 600 operation

The number of queueempty conditions are 24 after 600 operation

The number of queuefull conditions are 5 after 625 operation

The number of queueempty conditions are 24 after 625 operation

The number of queuefull conditions are 5 after 650 operation

The number of queueempty conditions are 24 after 650 operation

The number of queuefull conditions are 5 after 675 operation

The number of queueempty conditions are 24 after 675 operation

The number of queuefull conditions are 5 after 700 operation

The number of queueempty conditions are 24 after 700 operation

The number of queuefull conditions are 5 after 725 operation

The number of queueempty conditions are 24 after 725 operation

The number of queuefull conditions are 5 after 750 operation

The number of queueempty conditions are 24 after 750 operation

The number of queuefull conditions are 5 after 775 operation

The number of queueempty conditions are 24 after 775 operation

The number of queuefull conditions are 5 after 800 operation

The number of queueempty conditions are 24 after 800 operation

The number of queuefull conditions are 5 after 825 operation

The number of queueempty conditions are 24 after 825 operation

The number of queuefull conditions are 5 after 850 operation

The number of queueempty conditions are 24 after 850 operation

The number of queuefull conditions are 5 after 875 operation

The number of queueempty conditions are 24 after 875 operation

The number of queuefull conditions are 5 after 900 operation

The number of queueempty conditions are 24 after 900 operation

The number of queuefull conditions are 5 after 925 operation

The number of queueempty conditions are 24 after 925 operation

The number of queuefull conditions are 5 after 950 operation

The number of queueempty conditions are 24 after 950 operation

The number of queuefull conditions are 5 after 975 operation

The number of queueempty conditions are 24 after 975 operation

The number of queuefull conditions are 5 after 1000 operation

The number of queueempty conditions are 24 after 1000 operation

\*\*The number of queuefull and queueempty conditions of a circular queue after successive 25 operations\*\*

The number of queuefull conditions are 0 after 25 operation

The number of queueempty conditions are 1 after 25 operation

The number of queuefull conditions are 0 after 50 operation

The number of queueempty conditions are 4 after 50 operation

The number of queuefull conditions are 0 after 75 operation

The number of queueempty conditions are 7 after 75 operation

The number of queuefull conditions are 0 after 100 operation

The number of queueempty conditions are 9 after 100 operation

The number of queuefull conditions are 0 after 125 operation

The number of queueempty conditions are 9 after 125 operation

The number of queuefull conditions are 0 after 150 operation

The number of queueempty conditions are 14 after 150 operation

The number of queuefull conditions are 0 after 175 operation

The number of queueempty conditions are 14 after 175 operation

The number of queuefull conditions are 0 after 200 operation

The number of queueempty conditions are 14 after 200 operation

The number of queuefull conditions are 0 after 225 operation

The number of queueempty conditions are 14 after 225 operation

The number of queuefull conditions are 0 after 250 operation

The number of queueempty conditions are 18 after 250 operation

The number of queuefull conditions are 0 after 275 operation

The number of queueempty conditions are 25 after 275 operation

The number of queuefull conditions are 0 after 300 operation

The number of queueempty conditions are 25 after 300 operation

The number of queuefull conditions are 0 after 325 operation

The number of queueempty conditions are 26 after 325 operation

The number of queuefull conditions are 2 after 350 operation

The number of queueempty conditions are 26 after 350 operation

The number of queuefull conditions are 3 after 375 operation

The number of queueempty conditions are 26 after 375 operation

The number of queuefull conditions are 3 after 400 operation

The number of queueempty conditions are 26 after 400 operation

The number of queuefull conditions are 3 after 425 operation

The number of queueempty conditions are 26 after 425 operation

The number of queuefull conditions are 3 after 450 operation

The number of queueempty conditions are 27 after 450 operation

The number of queuefull conditions are 3 after 475 operation

The number of queueempty conditions are 31 after 475 operation

The number of queuefull conditions are 3 after 500 operation

The number of queueempty conditions are 36 after 500 operation

The number of queuefull conditions are 3 after 525 operation

The number of queueempty conditions are 36 after 525 operation

The number of queuefull conditions are 3 after 550 operation

The number of queueempty conditions are 36 after 550 operation

The number of queuefull conditions are 5 after 575 operation

The number of queueempty conditions are 36 after 575 operation

The number of queuefull conditions are 5 after 600 operation

The number of queueempty conditions are 43 after 600 operation

The number of queuefull conditions are 5 after 625 operation

The number of queueempty conditions are 45 after 625 operation

The number of queuefull conditions are 5 after 650 operation

The number of queueempty conditions are 45 after 650 operation

The number of queuefull conditions are 7 after 675 operation

The number of queueempty conditions are 45 after 675 operation

The number of queuefull conditions are 7 after 700 operation

The number of queueempty conditions are 45 after 700 operation

The number of queuefull conditions are 9 after 725 operation

The number of queueempty conditions are 45 after 725 operation

The number of queuefull conditions are 15 after 750 operation

The number of queueempty conditions are 45 after 750 operation

The number of queuefull conditions are 16 after 775 operation

The number of queueempty conditions are 45 after 775 operation

The number of queuefull conditions are 16 after 800 operation

The number of queueempty conditions are 45 after 800 operation

The number of queuefull conditions are 16 after 825 operation

The number of queueempty conditions are 45 after 825 operation

The number of queuefull conditions are 16 after 850 operation

The number of queueempty conditions are 45 after 850 operation

The number of queuefull conditions are 17 after 875 operation

The number of queueempty conditions are 45 after 875 operation

The number of queuefull conditions are 17 after 900 operation

The number of queueempty conditions are 45 after 900 operation

The number of queuefull conditions are 17 after 925 operation

The number of queueempty conditions are 47 after 925 operation

The number of queuefull conditions are 17 after 950 operation

The number of queueempty conditions are 47 after 950 operation

The number of queuefull conditions are 17 after 975 operation

The number of queueempty conditions are 47 after 975 operation

The number of queuefull conditions are 17 after 1000 operation

The number of queueempty conditions are 47 after 1000 operation

* **Proposed C Code**

**/\* ---------- queue.c--------------- \*/**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<time.h>**

**#define CIRCULAR\_QUEUE\_SIZE 16 /\* setting the size of circular queue as 16 \*/**

**int queuesize = 16; /\* initai size of normal queue is set as 16 \*/**

**int front = -1;/\* front,rear are set as -1 \*/**

**int rear = -1;**

**int fullcount = 0;/\* fullcount,emptycount counts the number of queuefull,queueempty conditions \*/**

**int emptycount = 0;**

**int queuefull = 0; /\* queuefull checks when normal queue is full and when the size is needed to be increased \*/**

**void QueueFull()**

**{**

**fullcount++; /\* QueueFull() function contains count of queuefull conditions \*/**

**}**

**void QueueEmpty()**

**{**

**emptycount++; /\* QueueEmpty() function contains count of queueempty conditions \*/**

**}**

**/\* function to enqueue \*/**

**void enQueue(int \*queue,int element)**

**{**

**if ( rear == queuesize-1 )**

**{**

**QueueFull();**

**queuefull=1;/\* when rear reaches the end then queuefull is set to 1 to denote the queuefull condition \*/**

**}**

**else**

**{**

**queue[++rear] = element; /\* enqueueing element \*/**

**}**

**}**

**/\* function to dequeue \*/**

**int deQueue(int \*queue)**

**{**

**if ( rear == front )**

**{**

**QueueEmpty();**

**return -1;**

**}**

**else**

**{**

**return queue[++front];/\* dequeueing \*/**

**}**

**}**

**/\* enqueueing for the circular queue \*/**

**void enCircularQueue(int \*circularqueue,int element)**

**{**

**if(front == -1 && rear == CIRCULAR\_QUEUE\_SIZE-1 )**

**{**

**QueueFull();/\* speacial case of queuefull condition \*/**

**}**

**/\* queuefull condition for circular queue \*/**

**else if ( front == (rear + 1)%CIRCULAR\_QUEUE\_SIZE )**

**{**

**QueueFull();**

**}**

**else**

**{**

**rear = (rear + 1)%CIRCULAR\_QUEUE\_SIZE;**

**circularqueue[rear] = element;/\* enqueueing \*/**

**}**

**}**

**/\* dequeueing for the circular queue \*/**

**int deCircularQueue(int \*circularqueue)**

**{**

**if ( rear == front )**

**{**

**QueueEmpty();**

**return -1;**

**}**

**else**

**{**

**front = (front + 1)%CIRCULAR\_QUEUE\_SIZE;**

**return circularqueue[front];/\* dequeueing \*/**

**}**

**}**

**int main()**

**{**

**int\* queue = malloc(queuesize\*sizeof(int));/\* formation of queue \*/**

**int\* circularqueue = malloc(CIRCULAR\_QUEUE\_SIZE\*sizeof(int));/\* formation of circular queue \*/**

**printf("\*\*\*\*\*The number of queuefull and queueempty conditions of a normal queue after successive 25 operations\*\*\*\*\*\n\n");**

**srand(time(0)); /\* srand is set with time(0) to generate different random numbers at different time \*/**

**int deletion = 0;/\* variable for storing dequeueing element \*/**

**for ( int i = 1 ; i <= 1000 ; i++ )**

**{**

**int random = rand()%2;/\* generating 0 or 1 as random number \*/**

**if ( random == 0 )**

**{**

**enQueue(queue,random);/\* 0 is chosen to enqueue \*/**

**/\* if quuefull is 1 i.e rear at the end so we need to increase the size for the current enqueue\*/**

**if ( queuefull == 1 )**

**{**

**queuesize = queuesize\*2;/\* applying growth strategy to increase the size \*/**

**int\* newqueue;**

**newqueue = (int\*)malloc(queuesize\*sizeof(int));**

**for(int i=0; i<=rear; i++)/\* copying the previous elements \*/**

**{**

**newqueue[i] = queue[i];**

**}**

**queue = newqueue;**

**queue[++rear] = random;/\* enqueueing the element after increased size \*/**

**queuefull = 0;/\* queuefull is again set to zero \*/**

**}**

**}**

**else**

**{**

**deletion = deQueue(queue);/\* if random = 1, dequeue option is chosen \*/**

**}**

**if ( i%25 == 0 )**

**{**

**/\* printing queuefull,queueempty condition after every 25 operations \*/**

**printf("\nThe number of queuefull conditions are %d after %d operation \n",fullcount,i);**

**printf("The number of queueempty conditions are %d after %d operation \n",emptycount,i);**

**}**

**}**

**fullcount = 0,emptycount = 0;/\* fullcount,emptycount are set to 0 for counting circular queue operations \*/**

**printf("\n\n\*\*The number of queuefull and queueempty conditions of a circular queue after successive 25 operations\*\*\n\n");**

**for ( int i = 1 ; i <= 1000 ; i++ )**

**{**

**int random = rand()%2;/\* generating 0 or 1 as random number \*/**

**if ( random == 0 )**

**{**

**enCircularQueue(circularqueue,random);/\* 0 is chosen to enqueue \*/**

**}**

**else**

**{**

**deletion = deCircularQueue(circularqueue);/\* if random = 1, dequeue option is chosen \*/**

**}**

**if ( i%25 == 0 )**

**{**

**/\* printing queuefull,queueempty condition after every 25 operations \*/**

**printf("\nThe number of queuefull conditions are %d after %d operation \n",fullcount,i);**

**printf("The number of queueempty conditions are %d after %d operation \n",emptycount,i);**

**}**

**}**

**free(queue); /\* freeing the used memory after desired output is generated \*/**

**free(circularqueue);**

**return 0;**

**}**

**/\*------------------------------------------------------------------------------------------------------------------------- \*/**

* **Conclusion**

**The proposed algorithm has overall runtime of O(n) where n is sum of the number of operations i.e 1000 .**

* **Limitations : The initial queue size of the normal queue is taken as 16. The size of the circular queue is fixed and taken as 16 which is less than 25 so that there is a chance of queuefull in the first 25 operations.**
* **Assumptions: For normal queue when the queue becomes full,the size is increased by growth strategy. For that reason,when the queue size is becoming so large then the chances of queuefull conditions become very small as compared to queue empty conditions.**