

LAB 5 : CIRCULAR QUEUE IMPLEMENTATION

WAP to simulate the working of a circular queue of integers using an array. Provide the following operations

a) Insert

b) Delete

c) Display

The program should print appropriate messages for queue empty and queue overflow conditions

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

int front=-1;
int rear=-1;

int queue[MAX];

void Enque(int);
int Deque();
void display();
int main(int argc, char **argv)
{
    int option;
    int item;
    do{
        printf("\n ----- \n");
        printf("\n Circular Queue \n");
        printf("\n 1. Insert to Queue ");
        printf("\n 2. delete from the Queue ");
        printf("\n 3. Display the content ");
        printf("\n 4. Exit \n");
        printf("Enter the option :");
        scanf("%d",&option);
        switch(option)
        {
            case 1: printf("Enter the element \n");
                    scanf("%d",&item);
                    Enque(item);
                    break;
            case 2: item=Deque();
                    if(item!=-1)
                        printf("Queue is empty \n");
                    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 ((rear+1)%MAX==front)
        printf("Queue is full\n");
    else
    {
        rear=(rear+1)%MAX;
        queue[rear]=ele;
        if(front ==-1)
            front=0;
    }
}

```

```

}
int Dequeue()
{
    int item;
    if((front == -1)&&(rear==-1))
        return -1;
    else
    {
        item=queue[front];
        front=(front+1)%MAX;
        if(front>rear)
        {
            front=-1;
            rear=-1;
        }
        return item;
    }
}

```

```

void display()

```

```

{
    int i;
    if((front==0)&& (rear==-1))
        printf("Queue is empty\n");
    else
    {
        printf("\n Queue contents:");
        for(i=front;i<=rear;i++)
            printf("\t %d", queue[i]);
    }
}

```

FOR **OUTPUT** I have define size as 3:

The screenshot displays the OnlineGDB beta IDE interface. On the left is a sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Sign Up, and Login. Below these are social media icons for Facebook, Twitter, and a '+ 71.7K' button. At the bottom of the sidebar is a 'GOT AN OPINION?' section with links to About, FAQ, Blog, Terms of Use, Contact Us, GDB Tutorial, Credits, and Privacy. The main area shows a C program for a Circular Queue. The program prompts the user to choose an option from a menu: 1. Insert to Queue, 2. delete from the Queue, 3. Display the content, 4. Exit. In the first execution, option 1 is selected and element 12 is entered. The output shows 'Circular Queue' followed by '12'. In the second execution, option 1 is selected again and element 16 is entered. The output shows 'Circular Queue' followed by '16'. The IDE title bar indicates the file is named 'input'.

```

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

f 71.7K

GOT AN OPINION?
SHARE AND GET REWARDED.
About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2020 GDB Online

input


Circular Queue

1. Insert to Queue
2. delete from the Queue
3. Display the content
4. Exit
Enter the option :1
Enter the element
12




Circular Queue


1. Insert to Queue
2. delete from the Queue
3. Display the content
4. Exit
Enter the option :1
Enter the element
16

```

 **OnlineGDB beta**
online compiler and debugger for c/c++
code. compile. run. debug. share.

[IDE](#)
[My Projects](#)
[Classroom new](#)
[Learn Programming](#)
[Programming Questions](#)
[Sign Up](#)
[Login](#)

 71.7K



GOT AN OPINION?
SHARE AND GET REWARDED.
[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)
© 2016 - 2020 GDB Online

input




```
-----  
Circular Queue  
1. Insert to Queue  
2. delete from the Queue  
3. Display the content  
4. Exit  
Enter the option :1  
Enter the element  
18  
-----  
Circular Queue  
1. Insert to Queue  
2. delete from the Queue  
3. Display the content  
4. Exit  
Enter the option :1  
Enter the element  
20  
Queue is full  
-----
```


Type here to search

11:12 AM
10/19/2020

 **OnlineGDB beta**
online compiler and debugger for c/c++
code. compile. run. debug. share.

[IDE](#)
[My Projects](#)
[Classroom new](#)
[Learn Programming](#)
[Programming Questions](#)
[Sign Up](#)
[Login](#)

 71.7K


GOT AN OPINION?
SHARE AND GET REWARDED.
[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)
© 2016 - 2020 GDB Online

input

```
-----  
Circular Queue  
1. Insert to Queue  
2. delete from the Queue  
3. Display the content  
4. Exit  
Enter the option :3  
  
Queue contents:      12      16      18  
-----  
Circular Queue  
1. Insert to Queue  
2. delete from the Queue  
3. Display the content  
4. Exit  
Enter the option :2  
Removed element from the queue 12  
-----  
Circular Queue  
1. Insert to Queue  
2. delete from the Queue
```

Type here to search

11:12 AM
10/19/2020

1BM19CS216