

DATE: PAGE:
3. b) write a program to simulate the working of the circular queue of integer using n integer provide the following operation
1) Insert, 2) delete 3) Display
The program should print queue empty, overflow condition

Algorithm (or) Pseudocode

1) Start

2) set front = -1

3) set Rear = -1

4) set size = N

5) ~~set~~ declare queue[size]

=> Enqueue (Insert)

If (front == 0 and Rear == size - 1)

(front == Rear + 1) then

print ("queue is full")

else

if front == -1 then

front = 0

endif

Rear = (Rear + 1) mod size

queue[Rear] = element

print "Inserted:", element

endif

=> Dequeue (Delete)

procedure Dequeue ()

if front == -1 then

print ("queue is empty")

else

print "deleted:", queue[front]

if front == Rear then

front = -1

Rear = -1

Else

front = (front + 1) %

Endif

check

⇒ Display

procedure Display()

if front == -1 then

print "Queue is empty"

else

print "Queue elements"

i = front

while True

print queue[i]

if i == Rear then

break

endif

i = (i+1) mod size

endwhile

endif

end procedure

3/11

#include <stdio.h>

#define n 100

int queue[n];

int front = -1;

int rear = -1;

void enqueue (int x)

{
if ((rear+1) % n == front) {
print ("Queue overflow\n");
}

else if (front == -1 & rear == -1) {

front = rear = 0;

queue[rear] = x;

}

DATE _____

```

else {
    rear = (rear + 1) % n;
    queue[rear] = x;
}
}

void dequeue() {
    if (front == -1 && rear == -1) {
        printf("Queue is empty\n");
    }
    else if (front == rear) {
        printf("Deleted element: %d\n", queue[front]);
        front = rear = -1;
    }
    else {
        printf("Deleted element: %d\n", queue[front]);
        front = (front + 1) % n;
    }
}

void display() {
    int i;
    if (front == -1 && rear == -1) {
        printf("Queue is empty\n");
    }
    else {
        printf("Queue elements:");
        i = front;
        while (i != rear) {
            printf("%d ", queue[i]);
            i = (i + 1) % n;
        }
        printf("%d\n", queue[rear]);
    }
}

```

```
void main() {  
    int ch;  
    while (1) {  
        printf("In queue operations: \n");  
        printf("1. enqueue \n");  
        printf("2. dequeue \n");  
        printf("3. display \n");  
        printf("Enter your choice:");  
        scanf("%d", &ch);  
        switch (ch) {  
            case 1:  
                printf("Enter a elements to insert:");  
                int x;  
                scanf("%d", &x);  
                enqueue(x);  
                break;  
            case 2:  
                dequeue();  
                break;  
            case 3:  
                display();  
                break;  
            default:  
                printf("\n invalid choice \n");  
                break;  
        }  
    }  
    return 0;  
}
```


Output: Queue operations:

1. enqueue

2. dequeue

3. display

Enter your choice: 1

Enter your elements to insert: 56

Enter your choice: 1

Enter your elements to insert: 113

Enter your choice: 1

Enter your elements to insert: 89

Enter your choice: 2

Enter your elements to insert: 2

Enter your choice: 2

Enter your elements to insert: 4

Queue flow

Enter your choice: 3

Queue elements: 56 113 89 2

Enter your choice: 2

Deleted element: 56

Enter your choice: 3

Queue elements: 113 89 2

Enter your choice: 2

Enter your elements to insert: 100

Enter your choice: 3

Queue element: 113 89 2 100

OP

3/11/20

e/p

```
circulqueue.c - CodeBlocks20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DsxyBlocks Settings Help
StartHere X circulqueue.c X
1 #include <stdio.h>
2 #define n 5
3
4 int queue[n];
5 int front = -1;
6 int rear = -1;
7
8 void enqueue(int x) {
9     if ((rear + 1) % n == front) {
10         printf("Queue Overflow\n");
11     }
12     else if (front == -1 && rear == -1) {
13         front = rear = 0;
14         queue[rear] = x;
15     }
16     else {
17         rear = (rear + 1) % n;
18         queue[rear] = x;
19     }
20 }
21
22 void dequeue() {
23     if (front == -1 && rear == -1) {
24         printf("Queue is empty\n");
25     }
26     else if (front == rear) {
27         printf("Deleted element: %d\n", queue[front]);
28         front = rear = -1;
29     }
30     else {
31         printf("Deleted element: %d\n", queue[front]);
32         front = (front + 1) % n;
33     }
34 }
35
36 void display() {
37     int i;
38     if (front == -1 && rear == -1) {
39         printf("Queue is empty\n");
40     }
41     else {
42         printf("Queue elements: ");
43         i = front;
44         while (i != rear) {
45             printf("%d ", queue[i]);
46             i = (i + 1) % n;
47         }
48         printf("%d\n", queue[rear]);
49     }
50 }
51
52 void main()
53 {
54     int ch;
55
56     while (1) {
57         printf("\nQueue Operations:\n");
58         printf("1. enqueue\n");
59         printf("2. dequeue\n");
60         printf("3. Display\n");
61         printf("Enter your choice: ");
62         scanf("%d", &ch);
63
64         switch (ch) {
65             case 1:
66                 printf("Enter a element to insert:");
67                 int x;
68                 scanf("%d", &x);
69                 enqueue(x);
70                 break;
71             case 2:
72                 dequeue();
73                 break;
74             case 3:
75                 display();
76                 break;
77             default:
78                 printf("Invalid choice\n");
79                 break;
80         }
81     }
82     return 0;
83 }
84
85 int main()
86 {
87     // ...
88 }
```

D:\chethan\CPP\circulqueue.c C/C++ Windows (CR-LF) WINDOWS-1252 Line 20, Col 2, Pos 369 Insert Read/Write default 09:23 03-10-2023

```
circulqueue.c - CodeBlocks20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Plugins DsxyBlocks Settings Help
StartHere X circulqueue.c X
38 if (front == -1 && rear == -1) {
39     printf("Queue is empty\n");
40 }
41 else {
42     printf("Queue elements: ");
43     i = front;
44     while (i != rear) {
45         printf("%d ", queue[i]);
46         i = (i + 1) % n;
47     }
48     printf("%d\n", queue[rear]);
49 }
50
51 void main()
52 {
53     int ch;
54
55     while (1) {
56         printf("\nQueue Operations:\n");
57         printf("1. enqueue\n");
58         printf("2. dequeue\n");
59         printf("3. Display\n");
60         printf("Enter your choice: ");
61         scanf("%d", &ch);
62
63         switch (ch) {
64             case 1:
65                 printf("Enter a element to insert:");
66                 int x;
67                 scanf("%d", &x);
68                 enqueue(x);
69                 break;
70             case 2:
71                 dequeue();
72                 break;
73             case 3:
74                 display();
75                 break;
76             default:
77                 printf("Invalid choice\n");
78                 break;
79         }
80     }
81     return 0;
82 }
83
84 int main()
85 {
86     // ...
87 }
```

D:\chethan\CPP\circulqueue.c C/C++ Windows (CR-LF) WINDOWS-1252 Line 20, Col 2, Pos 369 Insert Read/Write default 09:24 03-10-2023

```
Queue Operations:
1. enqueue
2. dequeue
3. Display
Enter your choice: 1
Enter a elements to Insert:56
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice: 1
Enter a elements to Insert:43
Queue Overflow
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice: 3
Queue elements: 2 23 24 56 43
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice: 2
Deleted element: 2
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice: 3
Queue elements: 23 24 56 43
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice: 1
Enter a elements to Insert:32
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice: 3
Queue elements: 23 24 56 43 32
Queue Operations:
1. enqueue
2. Dequeue
3. Display
Enter your choice:
```