

Program 4.

```
#include <stdio.h>
#include <stdlib.h>
#define S 3
int front = -1;
int rear = -1;
int queue[S];
void enqueue (int, int);
int dequeue (int);
void display (int);
int main (int argc, char** argv)
{
```

```
    int choice, SIZE;
```

```
    int item;
```

```
    printf ("Enter the size of queue \n");
    scanf ("%d", &SIZE);
```

```
    do {
```

```
        printf ("1. Insert in Queue ");
```

```
        printf ("2. Delete from Queue ");
```

```
        printf ("3. Display the content ");
```

```
        printf ("4. EXIT \n");
```

```
        printf ("Enter your choice : ");
```

```
        scanf ("%d", &choice);
```

```
        case 1 : if ((front == 0 && rear == SIZE-1) ||
                    (front == rear + 1))
                    break;
```

```
        printf ("Queue is full ");
        break;
```

```
        }
```

```
        printf ("Enter the element you want to insert
```

```
printf("%d", item);
enqueue(SIZE, item);
break;
```

```
case 2: item = Dequeue(SIZE);
```

```
if (item == -999)
```

```
printf("Queue is empty -- 1n");
```

```
else
```

```
printf("Removed element from queue  
%d\n", item);
```

```
break;
```

```
case 3: display(SIZE);
```

```
break;
```

```
case 4: printf("Exiting");
```

```
exit(0);
```

```
default: printf("INVALID CHOICE");
```

```
break;
```

```
}
```

```
} while (choice != 4);
```

```
return 0;
```

```
}
```

```
void Enqueue (int SIZE, int ele)
```

```
{
```

```
if ((front == 0 && rear == SIZE-1)
```

```
|| front == rear+1)
```

```
{
```

```
printf("Queue is full");
```

```
return;
```

```
} else {
```

```
rear = (rear + 1) % SIZE;
```

```
queue[rear] = ele;
```

```
if (front == -1)
```

```
front = 0;
```

```
}
```

```

int Dequeue (int SIZE) {
    int item;
    if ((front == -1) && (rear == -1))
        return (-999);
    else {
        item = queue[front];
        if (front == rear)
            front = -1;
            rear = -1;
        else {
            front = (front + 1) % SIZE;
        }
        return item;
    }
}

```

```

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

```