

Queues

1. Linear Queue

```
#include <stdio.h>
#define queue[N];
int front = -1, rear = -1;
void enqueue(int x)
{
    if (rear == N-1)
    {
        printf("overflow");
    }
    else
    {
        rear++;
        queue[rear] = x;
    }
}

void dequeue()
{
    if (front == -1 && rear == -1)
    {
        printf("underflow");
    }
    else
    {
        front++;
    }
}
```

```
printf("x.d", queue[front]);  
front++;
```

```
3  
void display()  
{
```

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

```
{  
printf("underflow");
```

```
}  
}
```

```
printf("Queue contains");
```

```
for (int i = front; i < rear + 1; i++)
```

```
{  
printf("x.d", queue[i]);
```

```
}
```

```
}
```

```
3
```

```
int main()
```

```
{
```

```
enqueue(10);
```

```
enqueue(20);
```

```
enqueue(30);
```

```
display queue;
```

```
int dequued = dequeue();
```



```
printf("Dequeued item is %d", dequeued);
display Queue();
return 0;
```

3

output:-

Queue contains

10

20

30

Queue contains

20

30

2. circular queue

```
#include <stdio.h>
```

```
#define N=5
```

```
int queue[N];
```

```
int front = -1, rear = -1, element;
```

```
void enqueue(int n)
```

```
{
```

```
for(front = -1; rear = -1;
```

```
{
```

```
front = rear = 0;
```

```
queue[rear] = n;
```

3
else if (rear+1) % N == front)

↙

printf("Queue is full");

3

else

rear = (rear+1) % N;

queue[rear] = x;

4

3

void dequeue()

↙

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

↙

printf("underflow");

3

else if (front == rear)

↙

~~front = rear = -1;~~

4

else

printf("%d", queue[front]);

front = (front+1) % N;


```
void display()
```

```
{
```

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

```
{
```

```
    printf("underflow")
```

```
}
```

```
else
```

```
    printf("Queue contains");
```

```
    int i = front;
```

```
    while (i != rear)
```

```
    {
```

```
        printf("%d", arr[i]);
```

```
        i = (i + 1) % N;
```

```
    }
```

```
    printf("%d", arr[rear]);
```

```
}
```

```
int main()
```

```
{
```

```
    enqueue(10);
```

```
    enqueue(20);
```

```
    enqueue(30);
```

```
    displayQueue();
```

```
    int dequued = dequeue();
```

```
    printf("Dequued element is %d", dequued);
```

```

display Queue();
enqueue(40);
enqueue(50);
display Queue();
return 0;

```

4

Output

Queue contains

10

20

30

Queue contains

10

30

Queue contains

20

30

40

50