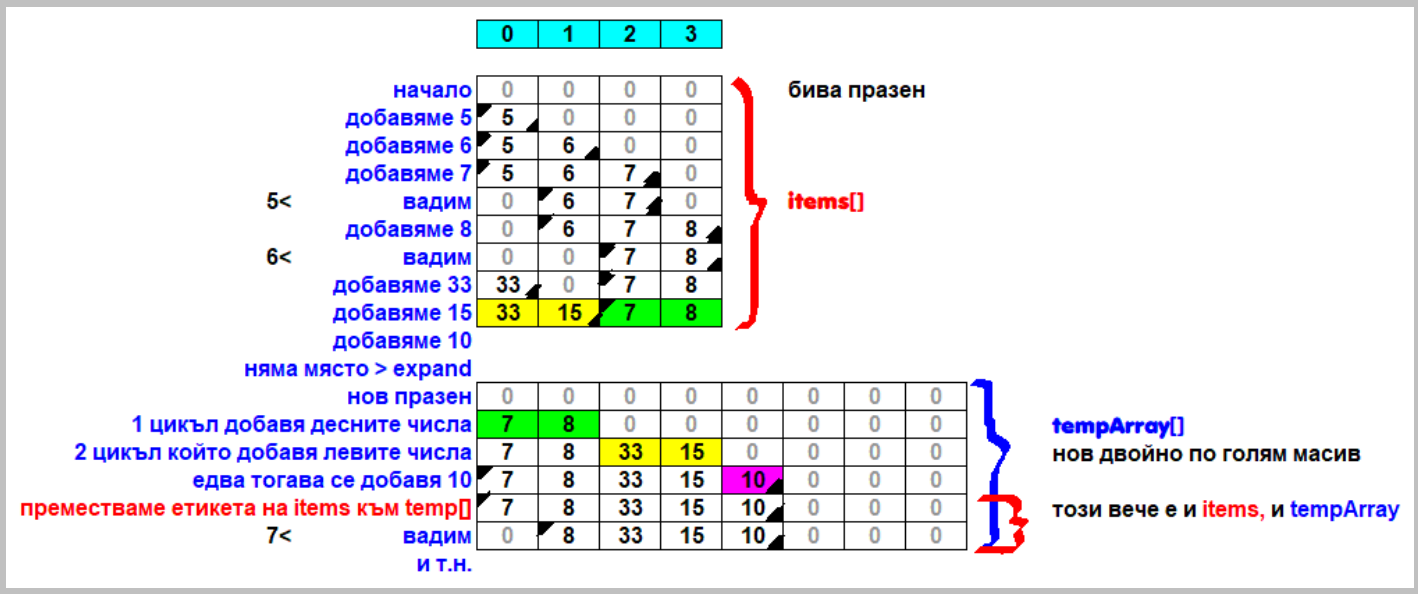
Как се случват нещата визуално при Опашка

реализирана с кръгов разширяем масив

- Горното ляво ъгълче показва елемента който е първи на опашката
- а долното дясно елемента който е последен в опашката




```
static void Main(string[] args)
{
    MyQueue queue = new MyQueue();

    queue.Enqueue(5); queue.ShowItems();
    queue.Enqueue(6); queue.ShowItems();
    queue.Enqueue(7); queue.ShowItems();

    int? x1 = queue.Dequeue(); queue.ShowItems(); //x1=5
    queue.Enqueue(8); queue.ShowItems();

    int? x2 = queue.Dequeue(); queue.ShowItems(); //x2=6
    queue.Enqueue(33); queue.ShowItems();
    queue.Enqueue(15); queue.ShowItems();
    queue.Enqueue(10); queue.ShowItems();

    int? x3 = queue.Dequeue(); queue.ShowItems(); //x2=6
}
```

```
Microsoft Visual Studio Debug Console
5 0 0 0
5 6 0 0
5 6 7 0
0 6 7 0
0 6 7 8
0 0 7 8
33 0 7 8
33 15 7 8
Разширявам капацитета до 8
7 8 33 15 10 0 0 0
0 8 33 15 10 0 0 0
```

Между другото > по препоръка на GPT (понеже му подхвърлих кода за да открие евентуални дупки) класическото решение при кръгови алгоритми е кода да бъде **endIndex = (endIndex + 1) % items.Length;** вместо проверка дали сме в края и **endIndex = -1,** което не е особено интуитивно и прави кода по-дълъг.