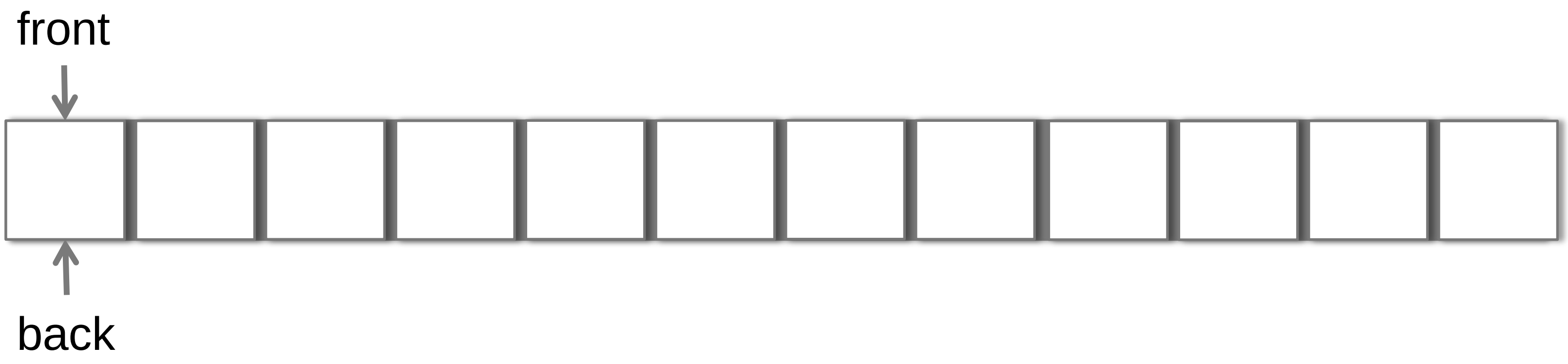
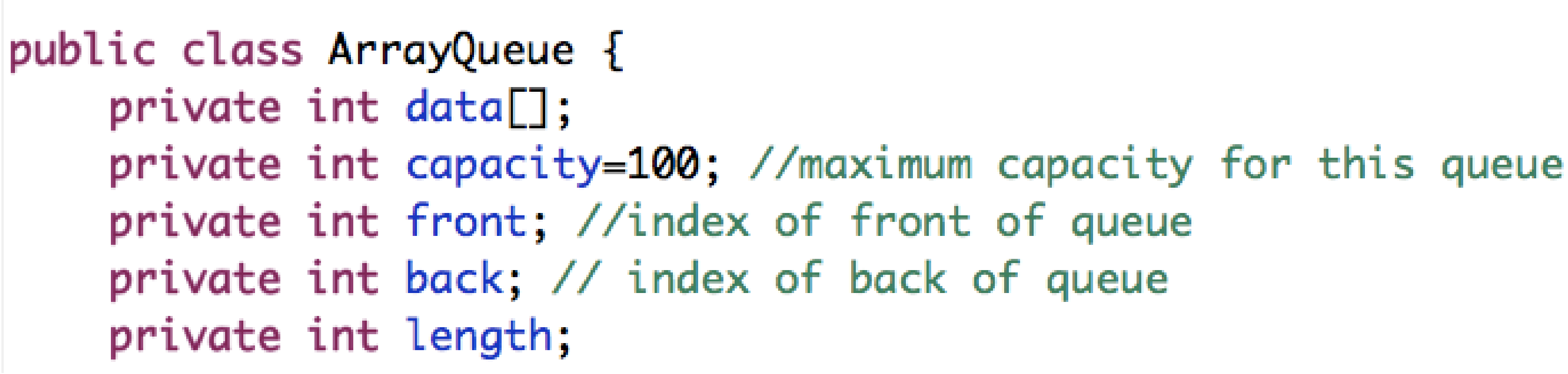
QUEUES & STACKS II

PART II : CIRCULAR QUEUE

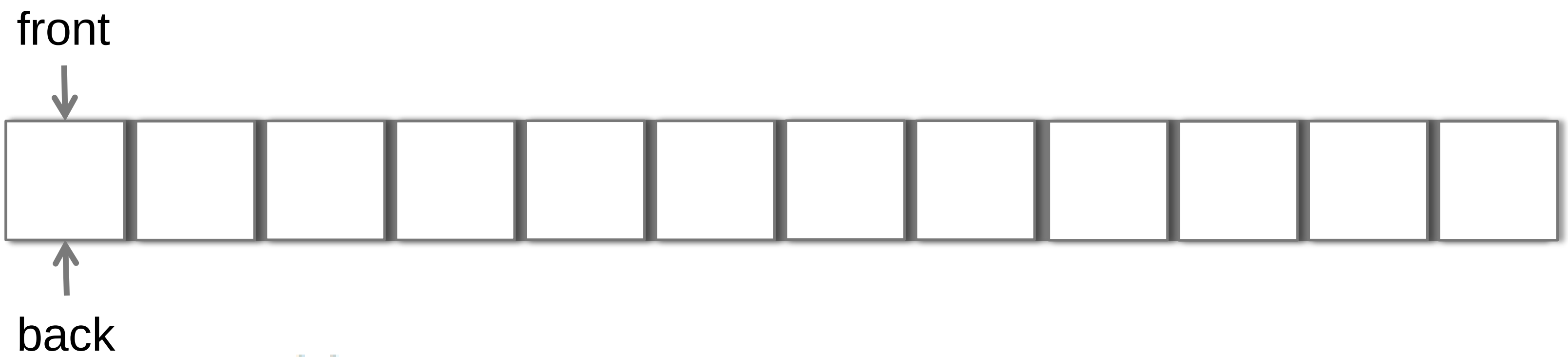
QUEUES
ARRAY IMPLEMENTATION





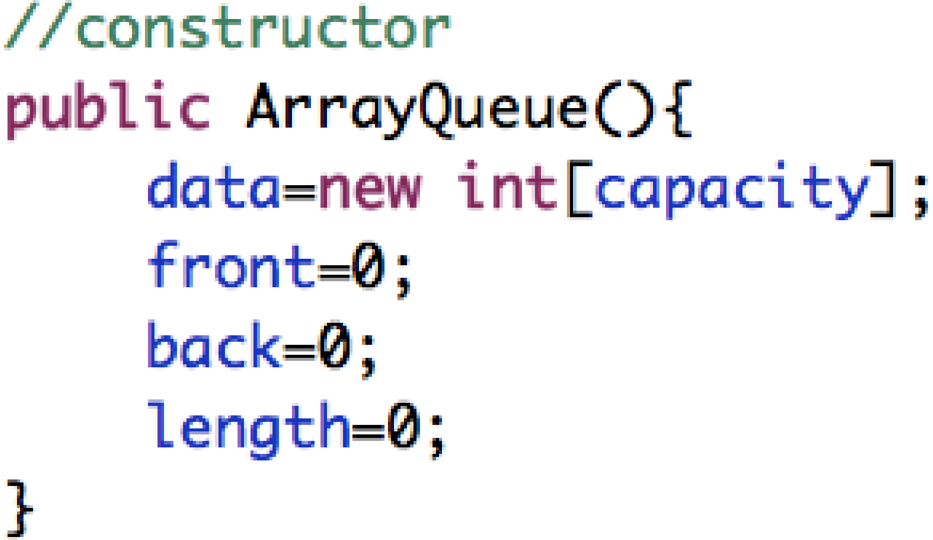
```
public class ArrayQueue {
    private int data[];
    private int capacity=100; //maximum capacity for this queue
    private int front; //index of front of queue
    private int back; // index of back of queue
    private int length;
```

QUEUES
ARRAY IMPLEMENTATION



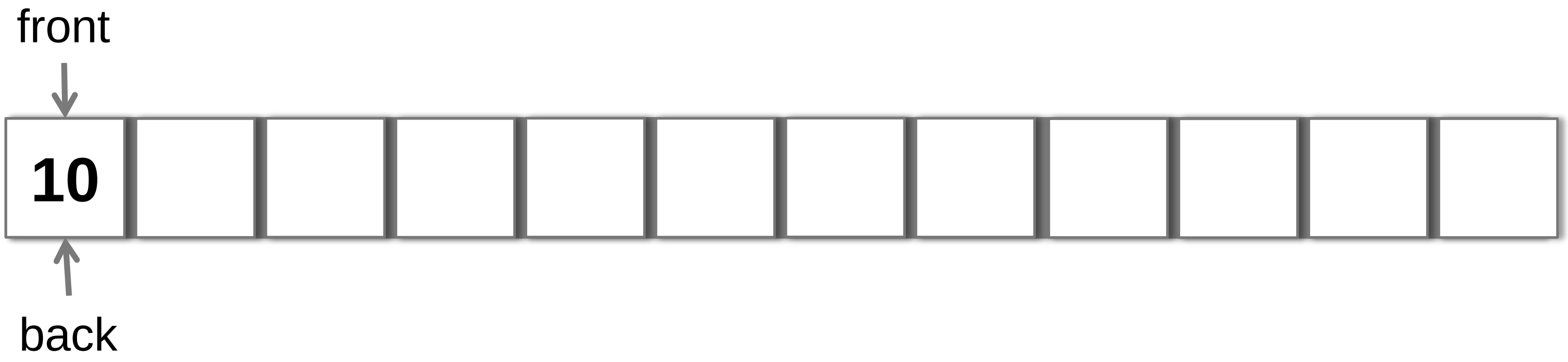
front

back



```
//constructor
public ArrayQueue(){
	data=new int[capacity];
	front=0;
	back=0;
	length=0;
}
```

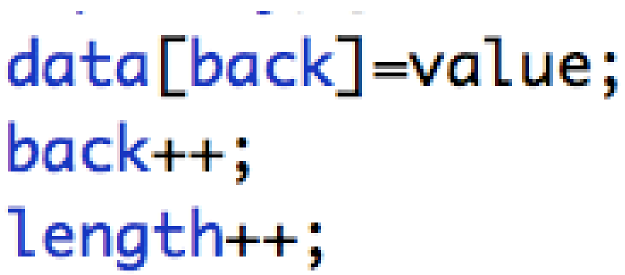
QUEUES
ARRAY IMPLEMENTATION



front

10

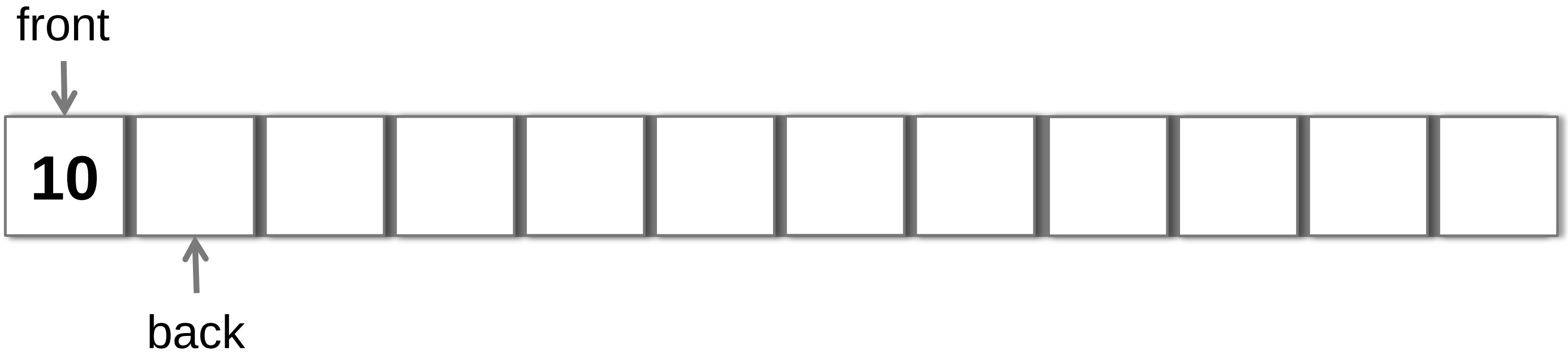
back

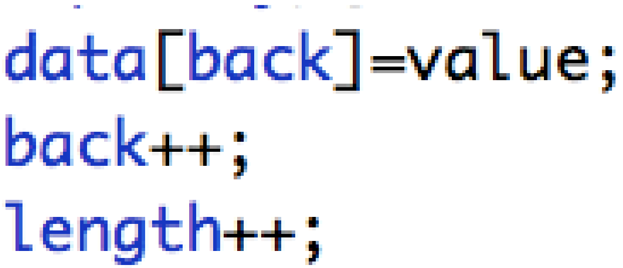


```
data[back]=value;
back++;
length++;
```

```
ArrayQueue q=new ArrayQueue();
q.add(10);
```

QUEUES
ARRAY IMPLEMENTATION

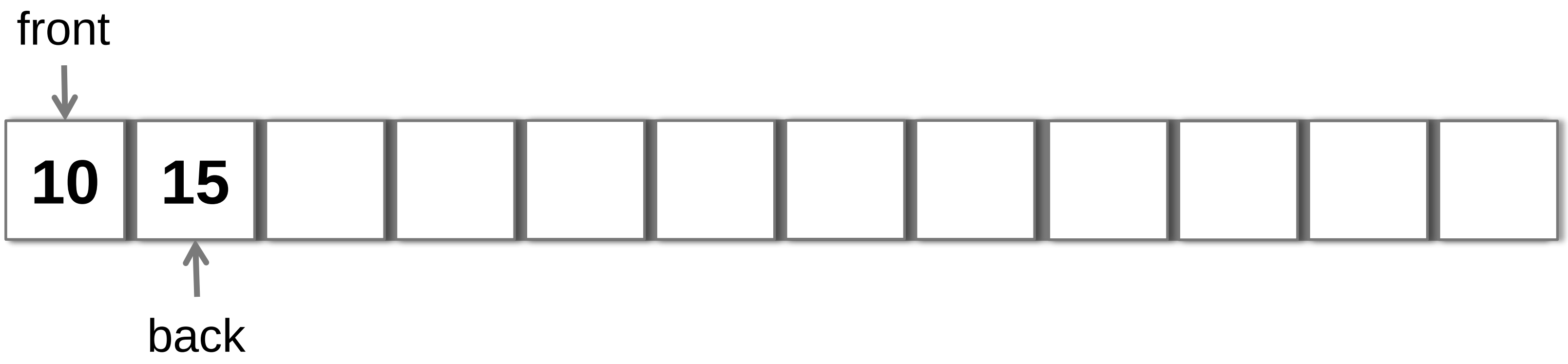


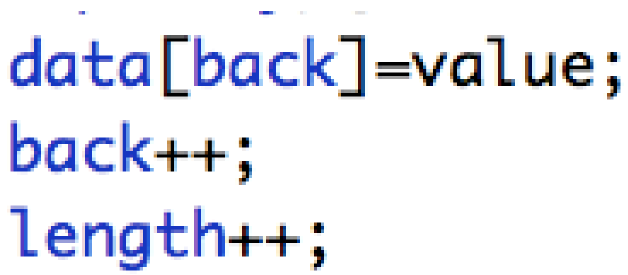


```
data[back]=value;
back++;
length++;
```

```
ArrayQueue q=new ArrayQueue();
q.add(10);
```

QUEUES
ARRAY IMPLEMENTATION

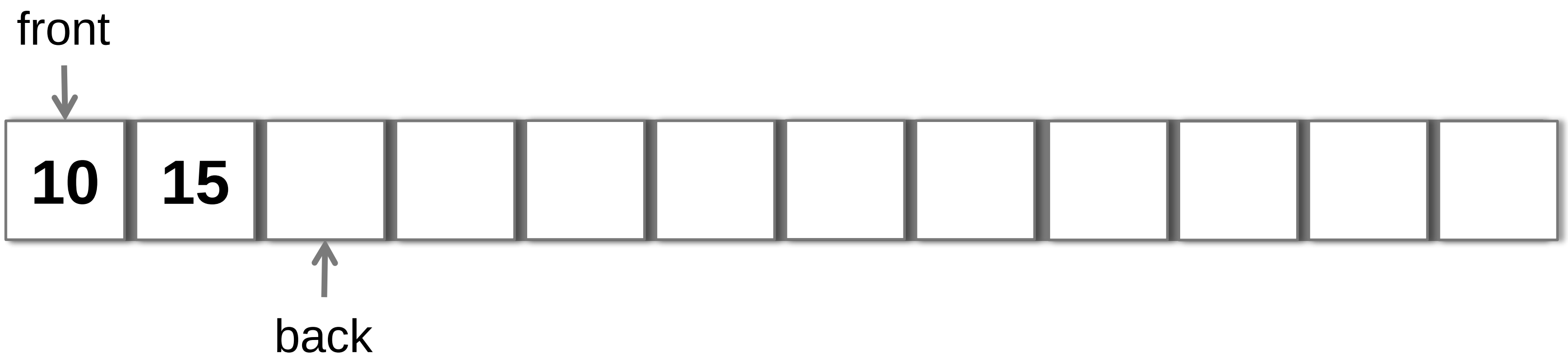


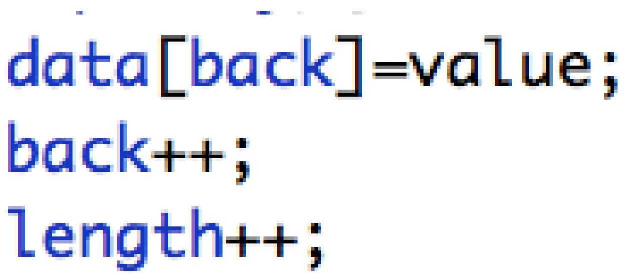


```
data[back]=value;
back++;
length++;
```

q.add(15);

QUEUES
ARRAY IMPLEMENTATION

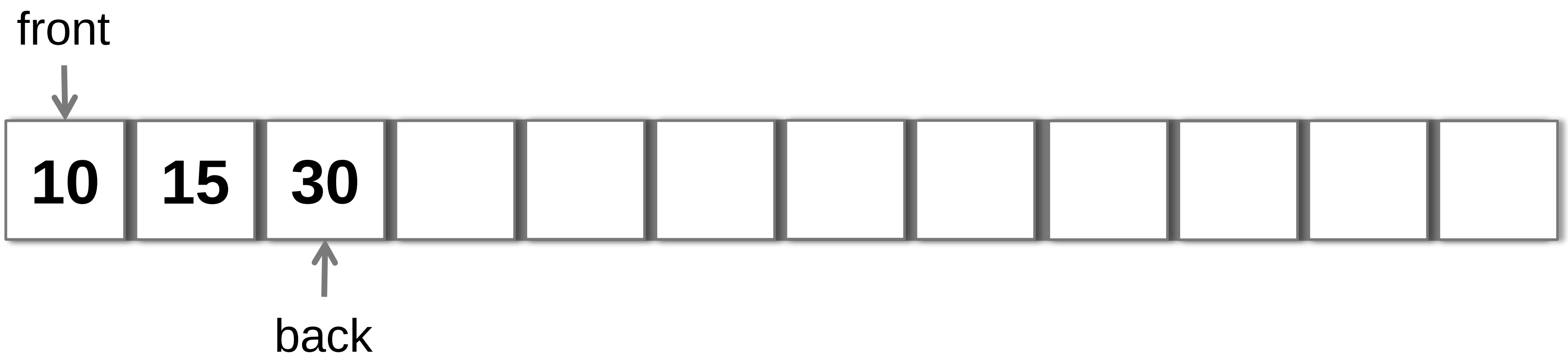




```
data[back]=value;
back++;
length++;
```

q.add(15);

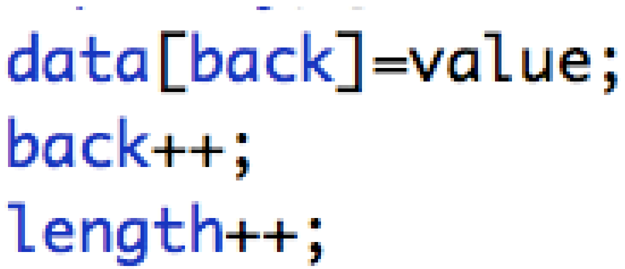
QUEUES
ARRAY IMPLEMENTATION



front

10	15	30									

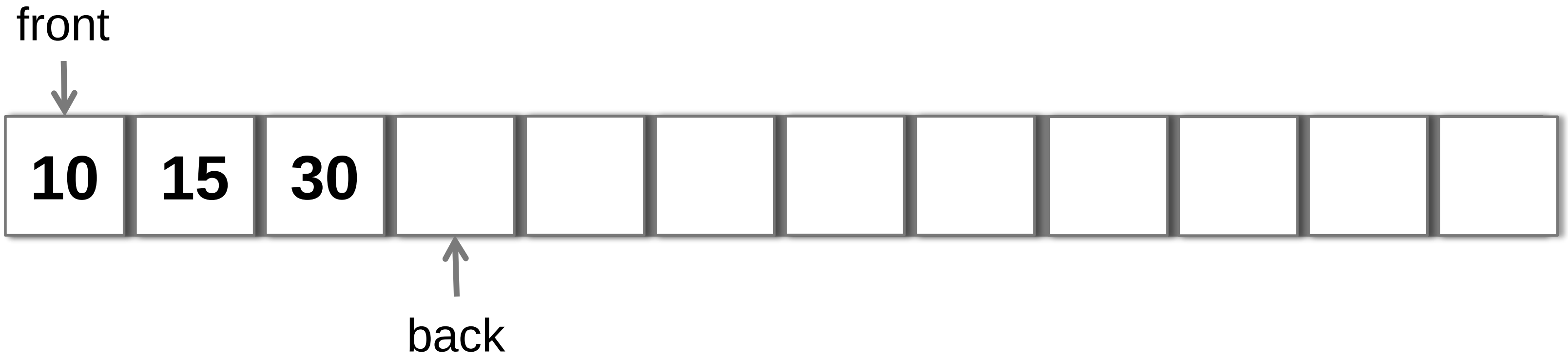
back

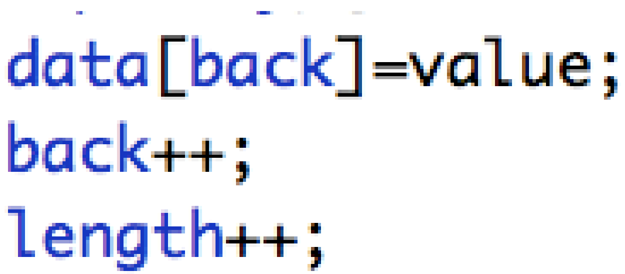


```
data[back]=value;
back++;
length++;
```

q.add(30);

QUEUES
ARRAY IMPLEMENTATION

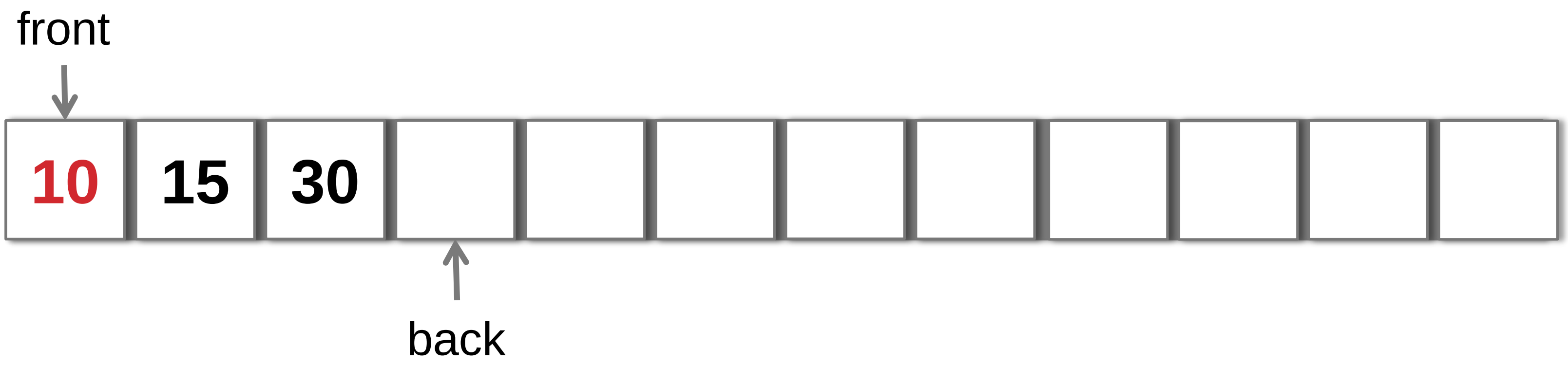




```
data[back]=value;
back++;
length++;
```

q.add(30);

QUEUES
ARRAY IMPLEMENTATION



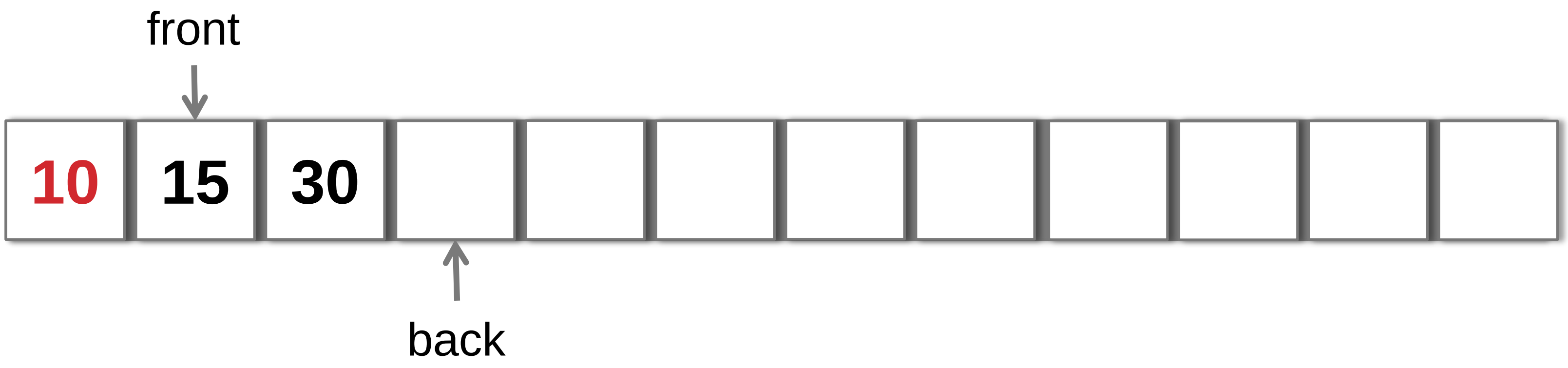
```
q.poll();
```

Poll()

Return the element at the front index

Move front index

QUEUES
ARRAY IMPLEMENTATION



front

10 | 15 | 30

back

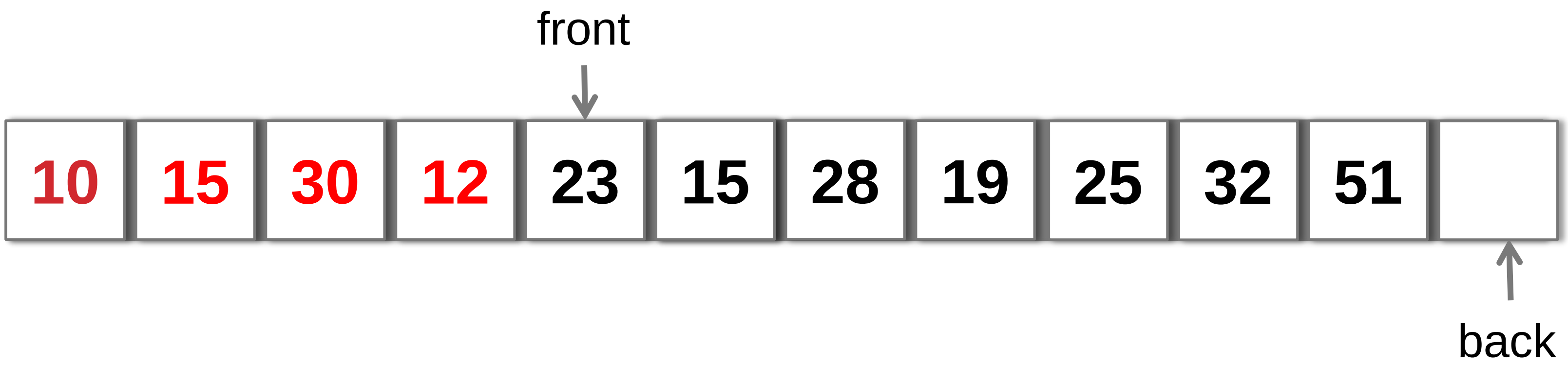
```
q.poll();
```

Poll()

Return the element at the front index

Move front index

QUEUES
ARRAY IMPLEMENTATION



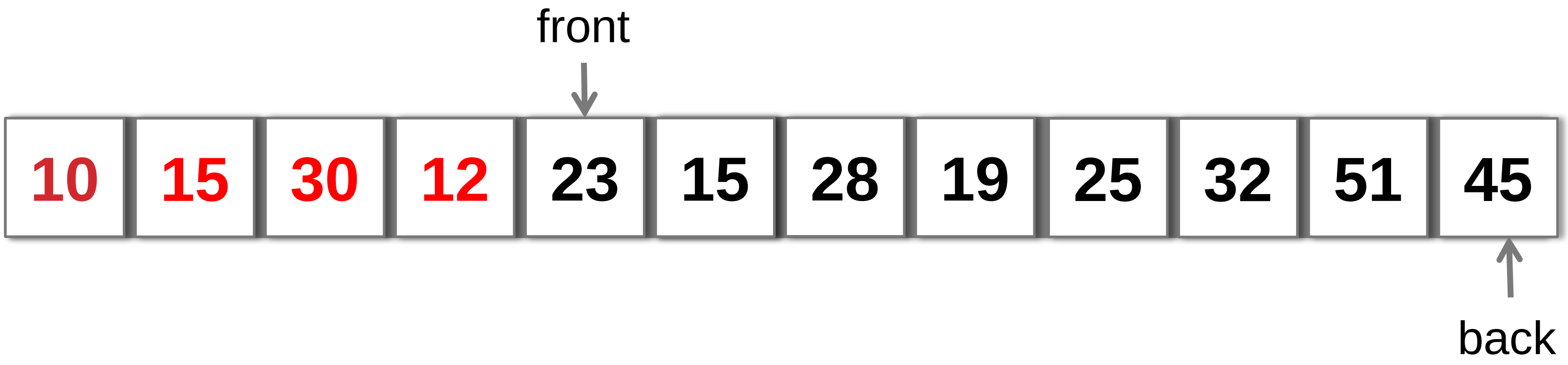
front

10	15	30	12	23	15	28	19	25	32	51	

back

After several add() and poll() calls …..

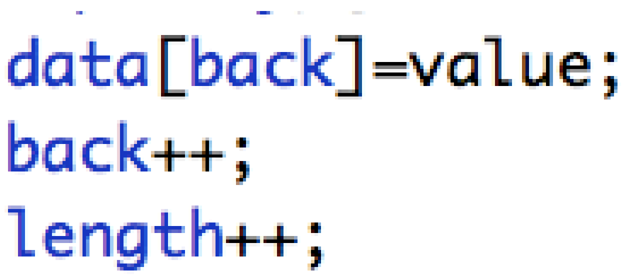
QUEUES
ARRAY IMPLEMENTATION



front

10	15	30	12	23	15	28	19	25	32	51	45

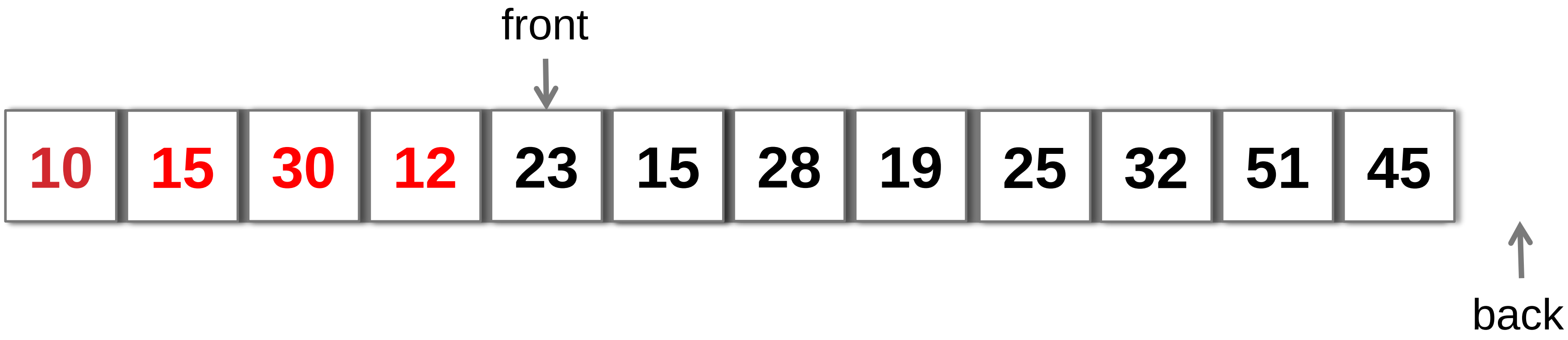
back



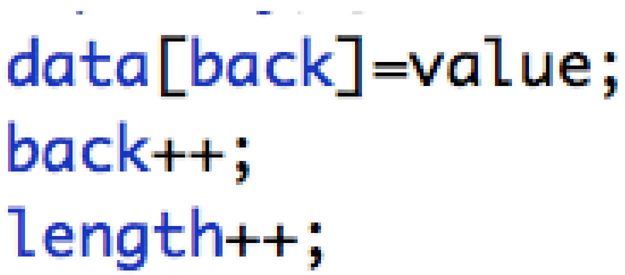
```
data[back]=value;
back++;
length++;
```

q.add(45);

QUEUES
ARRAY IMPLEMENTATION



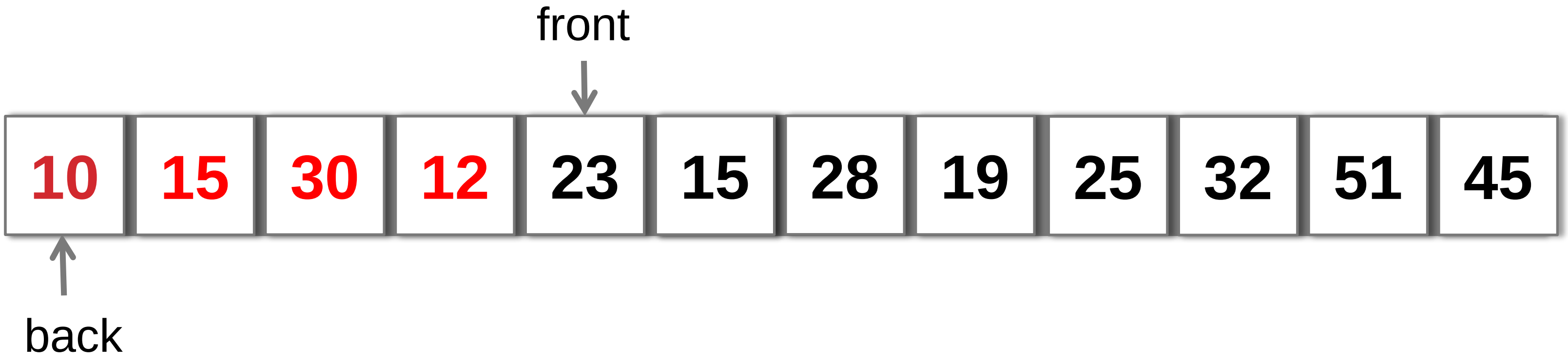
10	15	30	12	23	15	28	19	25	32	51	45



```
data[back]=value;
back++;
length++;
```

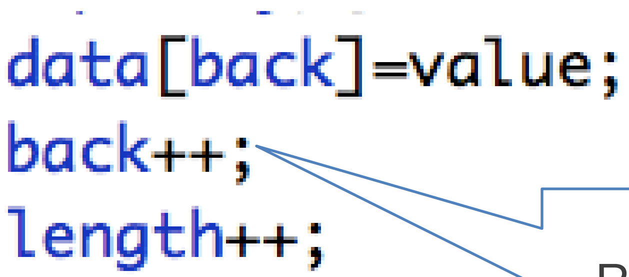
q.add(45);

QUEUES
ARRAY IMPLEMENTATION



10	15	30	12	23	15	28	19	25	32	51	45

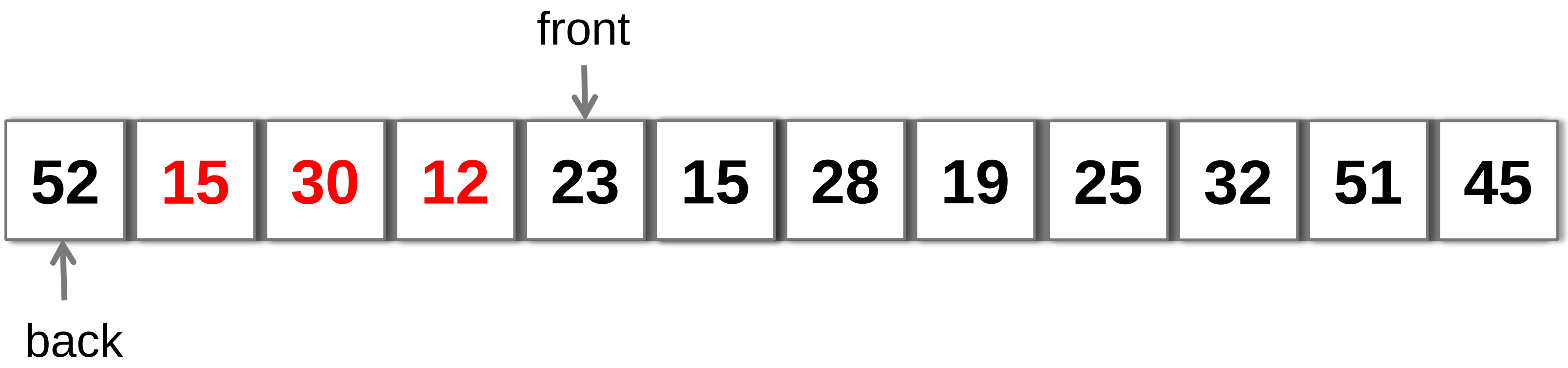
q.add(45);



```
data[back]=value;
back++;
length++;
```

Reset back to 0 when it reaches the end of the array

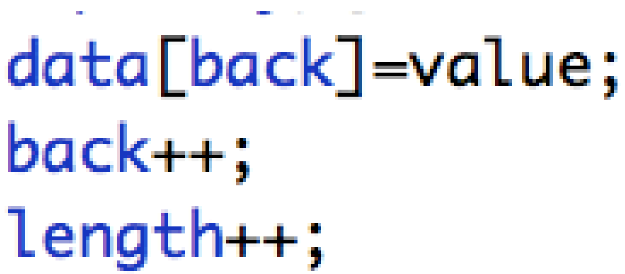
QUEUES
ARRAY IMPLEMENTATION



front

52	15	30	12	23	15	28	19	25	32	51	45

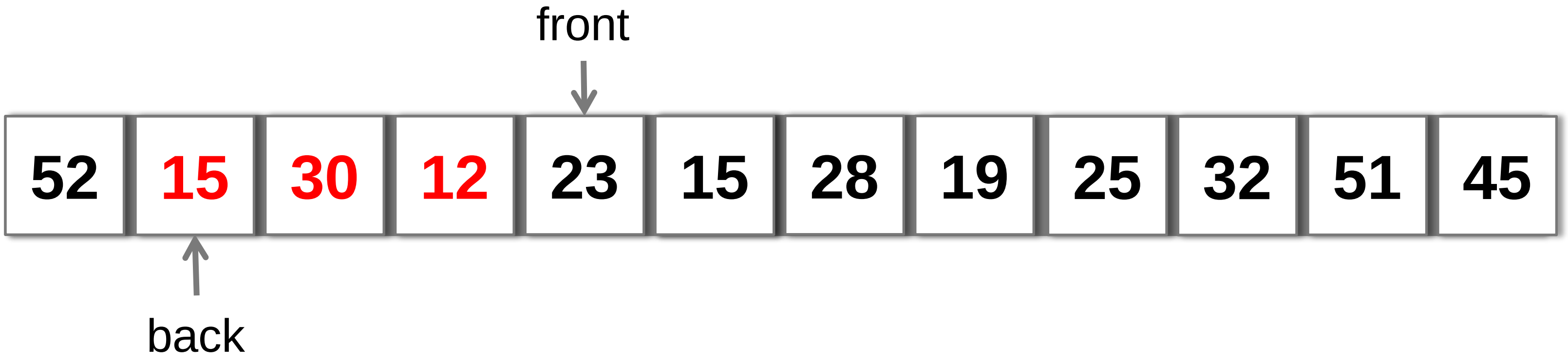
back



```
data[back]=value;
back++;
length++;
```

q.add(52);

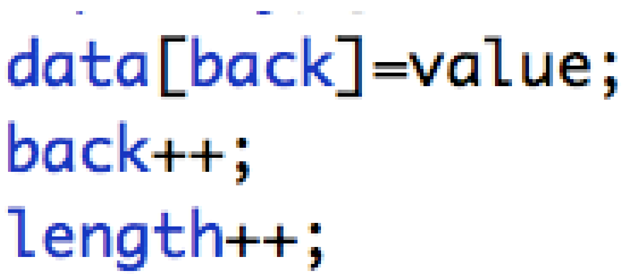
QUEUES
ARRAY IMPLEMENTATION



front

52	15	30	12	23	15	28	19	25	32	51	45

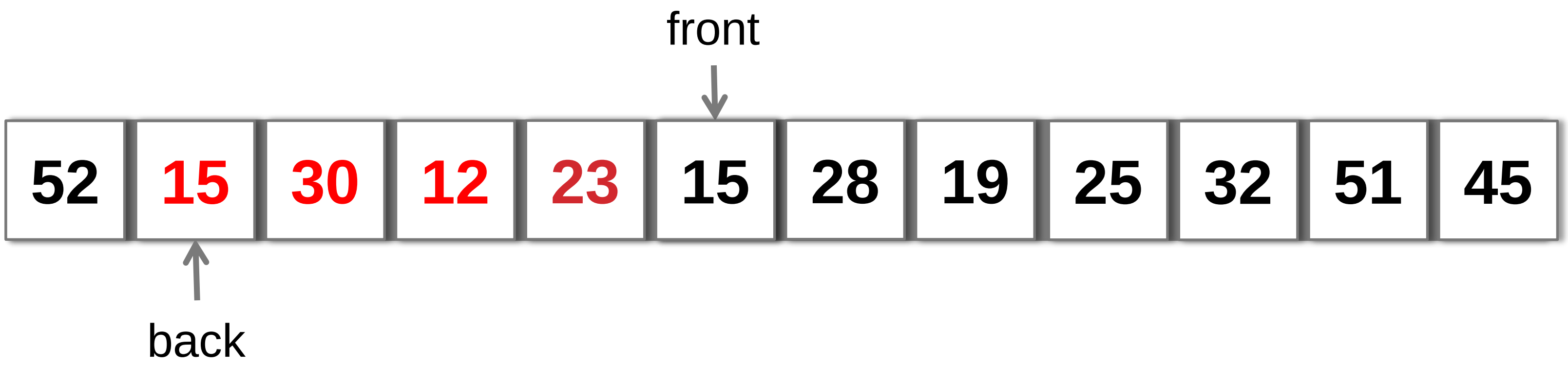
back



```
data[back]=value;
back++;
length++;
```

q.add(52);

QUEUES
ARRAY IMPLEMENTATION



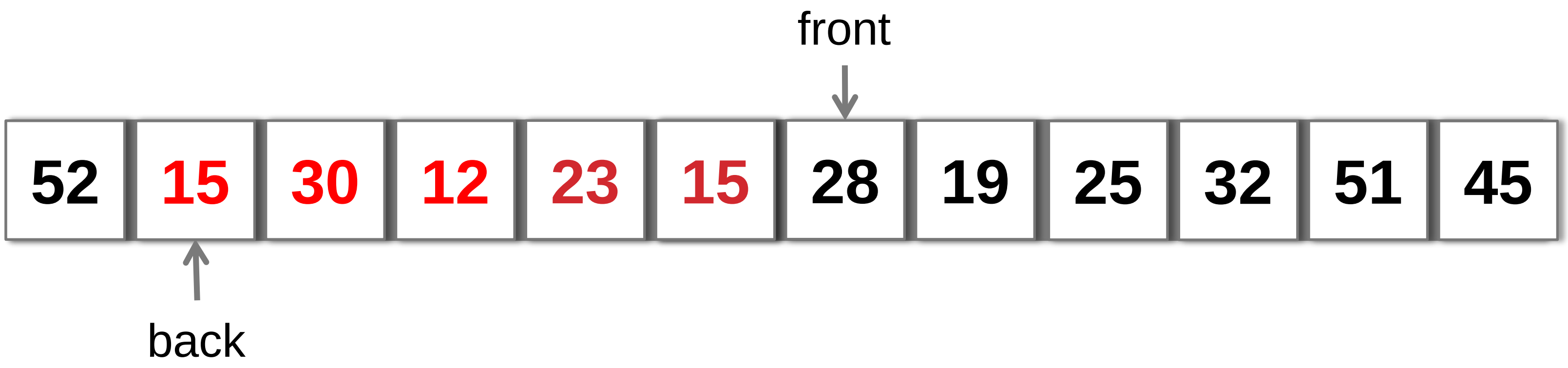
front

52	15	30	12	23	15	28	19	25	32	51	45

back

```
q.poll();
```

QUEUES
ARRAY IMPLEMENTATION



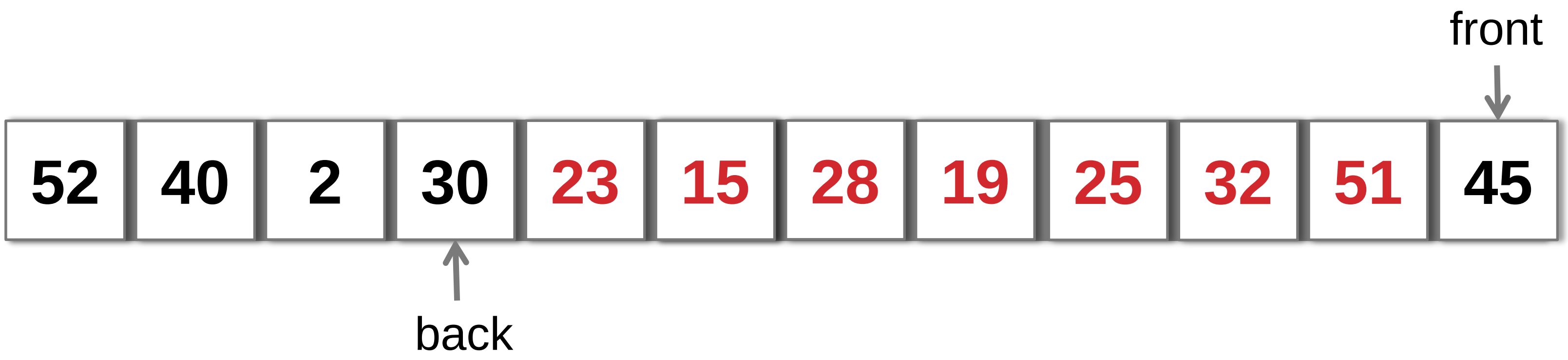
front

52	15	30	12	23	15	28	19	25	32	51	45

back

```
q.poll();
```

QUEUES
ARRAY IMPLEMENTATION



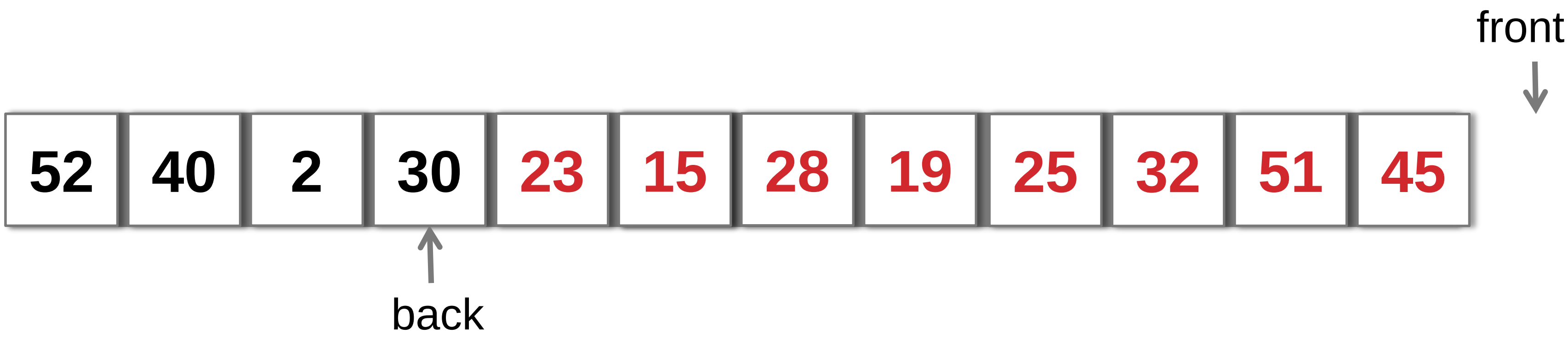
front

52	40	2	30	23	15	28	19	25	32	51	45

back

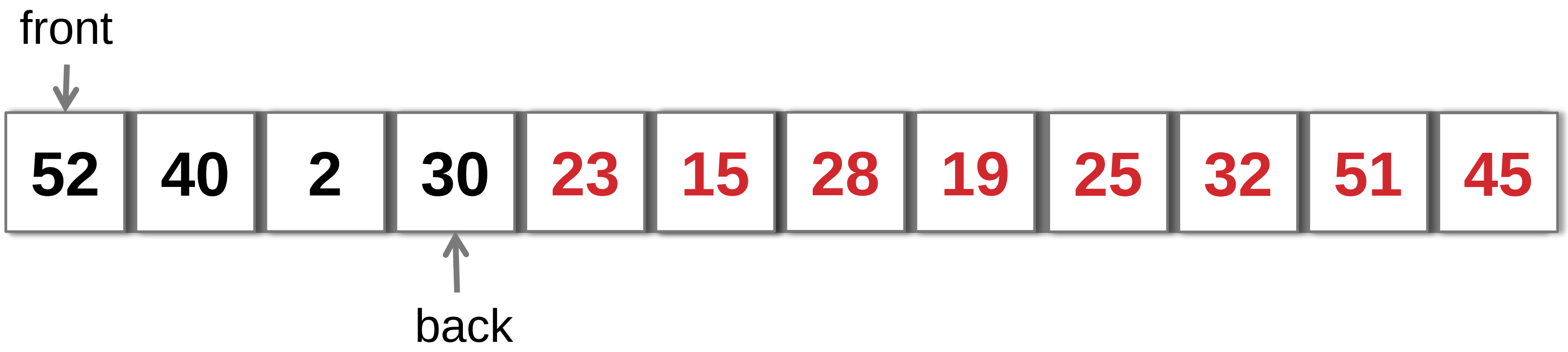
```
q.poll();
```

QUEUES
ARRAY IMPLEMENTATION



```
q.poll();
```

QUEUES
ARRAY IMPLEMENTATION



front

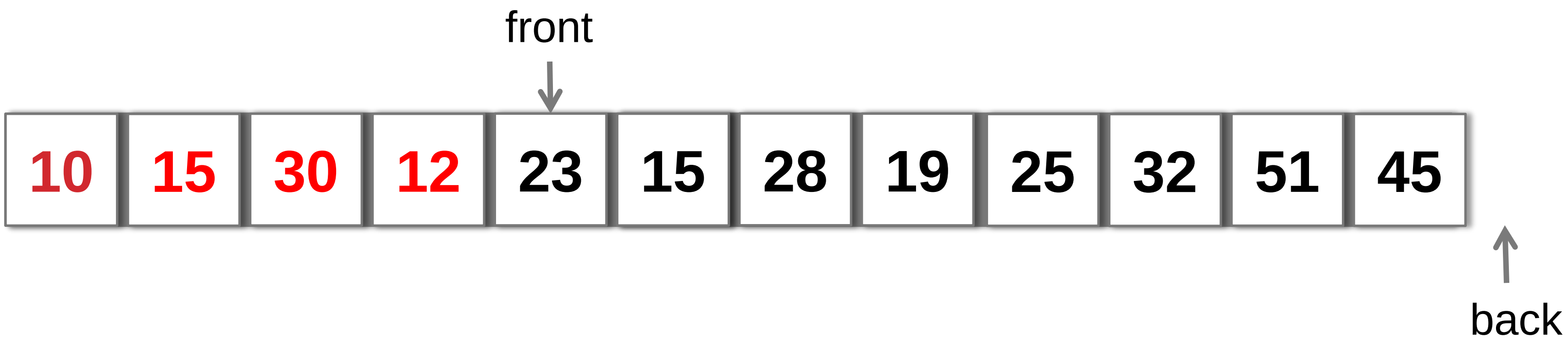
52	40	2	30	23	15	28	19	25	32	51	45

back

```
q.poll();
```

Reset front to index 0 when it reaches the end of the array

QUEUES
ARRAY IMPLEMENTATION



front

10	15	30	12	23	15	28	19	25	32	51	45

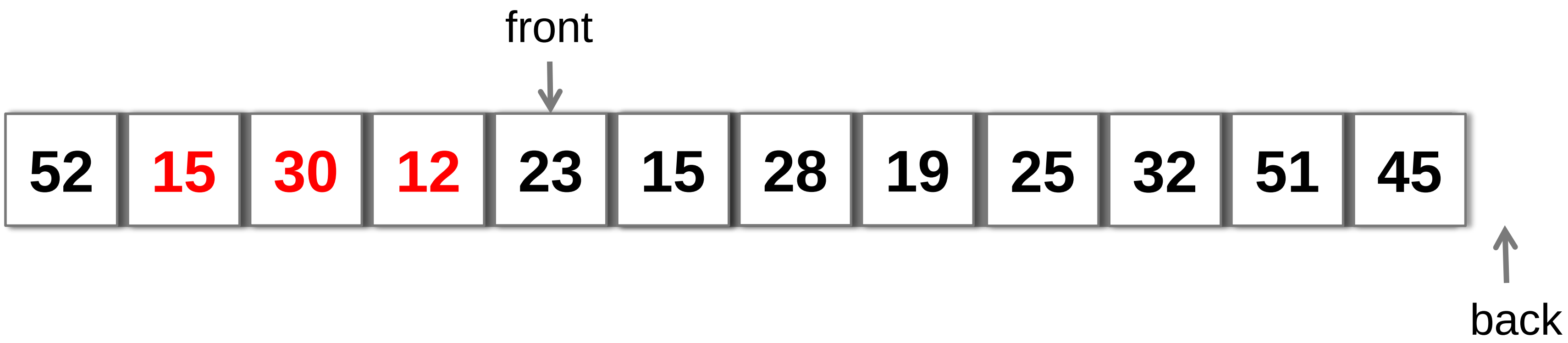
back

Capacity = 12
Array indices : 0 to 11
Back =12

back % capacity = 12 % 12 = 0

```
q.add(52);
```

QUEUES
ARRAY IMPLEMENTATION



front

52	**15**	**30**	**12**	**23**	**15**	**28**	**19**	**25**	**32**	**51**	**45**

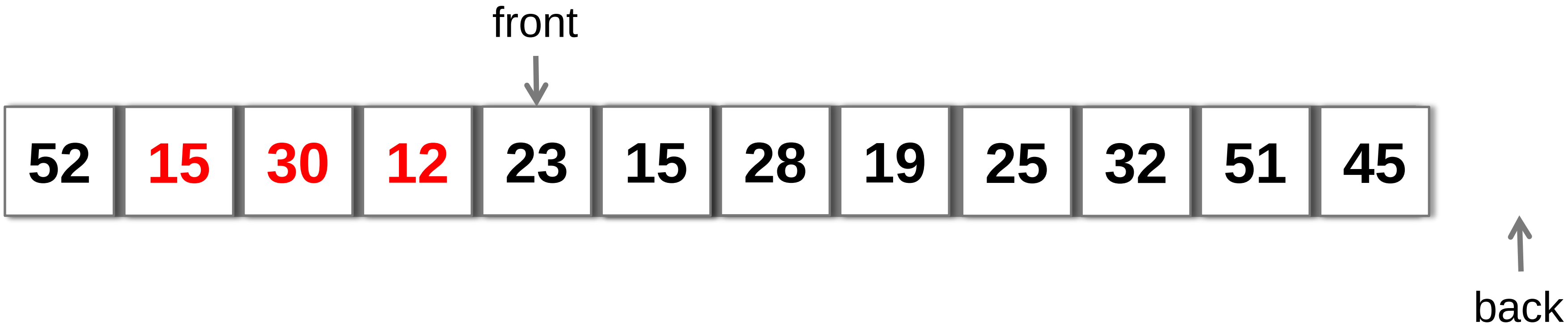
back

Capacity = 12
Array indices : 0 to 11
Back =12

back % capacity = 12 % 12 = 0

q.add(52);

QUEUES
ARRAY IMPLEMENTATION



front

52	15	30	12	23	15	28	19	25	32	51	45

back

Capacity = 12
Array indices : 0 to 11
Back =12

back % capacity = 12 % 12 = 0

q.add(52);