Java

```java
class MRUQueue {

    public MRUQueue(int n) {

    }

    public int fetch(int k) {

    }
}

/**
 * Your MRUQueue object will be instantiated and called as such:
 * MRUQueue obj = new MRUQueue(n);
 * int param_1 = obj.fetch(k);
 */
```

JavaScript

```javascript
/**
 * @param {number} n
 */
var MRUQueue = function(n) {

};

/**
 * @param {number} k
 * @return {number}
```

```
 */
MRUQueue.prototype.fetch = function(k) {

};

/**
 * Your MRUQueue object will be instantiated and called as such:
 * var obj = new MRUQueue(n)
 * var param_1 = obj.fetch(k)
 */
```

TypeScript

```
class MRUQueue {
    constructor(n: number) {

    }

    fetch(k: number): number {

    }
}

/**
 * Your MRUQueue object will be instantiated and called as such:
 * var obj = new MRUQueue(n)
 * var param_1 = obj.fetch(k)
 */
```

C++

```cpp
class MRUQueue {
public:
    MRUQueue(int n) {

    }

    int fetch(int k) {

    }
};

/**
 * Your MRUQueue object will be instantiated and called as such:
 * MRUQueue* obj = new MRUQueue(n);
 * int param_1 = obj->fetch(k);
 */
```

C#

```csharp
public class MRUQueue {

    public MRUQueue(int n) {

    }

    public int Fetch(int k) {

    }
}
```

```
/**
 * Your MRUQueue object will be instantiated and called as such:
 * MRUQueue obj = new MRUQueue(n);
 * int param_1 = obj.Fetch(k);
 */
```

Kotlin

```
class MRUQueue(n: Int) {

    fun fetch(k: Int): Int {

    }

}

/**
 * Your MRUQueue object will be instantiated and called as such:
 * var obj = MRUQueue(n)
 * var param_1 = obj.fetch(k)
 */
```

Go

```
type MRUQueue struct {

}
```

```go
func Constructor(n int) MRUQueue {

}


func (this *MRUQueue) Fetch(k int) int {

}


/**
 * Your MRUQueue object will be instantiated and called as such:
 * obj := Constructor(n);
 * param_1 := obj.Fetch(k);
 */
```
