Java

```java
class Solution {
    public int[] processQueries(int c, int[][] connections, int[][] queries) {

    }
}
```

JavaScript

```javascript
/**
 * @param {number} c
 * @param {number[][]} connections
 * @param {number[][]} queries
 * @return {number[]}
 */
var processQueries = function(c, connections, queries) {

};
```

TypeScript

```typescript
function processQueries(c: number, connections: number[][], queries: number[][]): number[] {

};
```

C++

```cpp
class Solution {
public:
    vector<int> processQueries(int c, vector<vector<int>>& connections, vector<vector<int>>& queries) {

    }
};
```

C#

```cs
public class Solution {
    public int[] ProcessQueries(int c, int[][] connections, int[][] queries) {

    }
}
```

Kotlin

```kt
class Solution {
    fun processQueries(c: Int, connections: Array<IntArray>, queries: Array<IntArray>): IntArray {

    }
}
```

Go

```go
func processQueries(c int, connections [][]int, queries [][]int) []int {

}
```
