

Java

```
class Encrypter {  
    public Encrypter(char[] keys, String[] values, String[] dictionary) {  
    }  
  
    public String encrypt(String word1) {  
    }  
  
    public int decrypt(String word2) {  
    }  
}  
  
/**  
 * Your Encrypter object will be instantiated and called as such:  
 * Encrypter obj = new Encrypter(keys, values, dictionary);  
 * String param_1 = obj.encrypt(word1);  
 * int param_2 = obj.decrypt(word2);  
 */
```

JavaScript

```
/**  
 * @param {character[]} keys  
 * @param {string[]} values  
 * @param {string[]} dictionary  
 */
```

```

var Encrypter = function(keys, values, dictionary) {

};

/**
 * @param {string} word1
 * @return {string}
 */
Encrypter.prototype.encrypt = function(word1) {

};

/**
 * @param {string} word2
 * @return {number}
 */
Encrypter.prototype.decrypt = function(word2) {

};

/**
 * Your Encrypter object will be instantiated and called as such:
 * var obj = new Encrypter(keys, values, dictionary)
 * var param_1 = obj.encrypt(word1)
 * var param_2 = obj.decrypt(word2)
 */

```

C++

```

class Encrypter {
public:

```

```

    Encrypter(vector<char>& keys, vector<string>& values, vector<string>& dictionary) {

    }

    string encrypt(string word1) {

    }

    int decrypt(string word2) {

    }
};

/**
 * Your Encrypter object will be instantiated and called as such:
 * Encrypter* obj = new Encrypter(keys, values, dictionary);
 * string param_1 = obj->encrypt(word1);
 * int param_2 = obj->decrypt(word2);
 */

```

C#

```

public class Encrypter {

    public Encrypter(char[] keys, string[] values, string[] dictionary) {

    }

    public string Encrypt(string word1) {

    }

}

```

```
    public int Decrypt(string word2) {  
    }  
}
```

```
/**  
 * Your Encrypter object will be instantiated and called as such:  
 * Encrypter obj = new Encrypter(keys, values, dictionary);  
 * string param_1 = obj.Encrypt(word1);  
 * int param_2 = obj.Decrypt(word2);  
 */
```
