

Java

```
public class Codec {  
  
    // Encodes a list of strings to a single string.  
    public String encode(List<String> strs) {  
  
    }  
  
    // Decodes a single string to a list of strings.  
    public List<String> decode(String s) {  
  
    }  
}  
  
// Your Codec object will be instantiated and called as such:  
// Codec codec = new Codec();  
// codec.decode(codec.encode(strs));
```

JavaScript

```
/**  
 * Encodes a list of strings to a single string.  
 *  
 * @param {string[]} strs  
 * @return {string}  
 */  
var encode = function(strs) {  
  
};
```

```
/**  
 * Decodes a single string to a list of strings.  
 *  
 * @param {string} s  
 * @return {string[]}  
 */  
var decode = function(s) {  
};  
  
/**  
 * Your functions will be called as such:  
 * decode(encode(strs));  
 */
```

TypeScript

```
/**  
 * Encodes a list of strings to a single string.  
 */  
function encode(strs: string[]): string {  
};  
  
/**  
 * Decodes a single string to a list of strings.  
 */  
function decode(s: string): string[] {  
};
```

```
/**  
 * Your functions will be called as such:  
 * decode(encode(strs));  
 */
```

C++

```
class Codec {  
public:  
  
    // Encodes a list of strings to a single string.  
    string encode(vector<string>& strs) {  
  
    }  
  
    // Decodes a single string to a list of strings.  
    vector<string> decode(string s) {  
  
    }  
};  
  
// Your Codec object will be instantiated and called as such:  
// Codec codec;  
// codec.decode(codec.encode(strs));
```

C#

```
public class Codec {  
  
    // Encodes a list of strings to a single string.
```

```
public string encode(IList<string> strs) {  
}  
  
// Decodes a single string to a list of strings.  
public IList<string> decode(string s) {  
}  
}  
  
// Your Codec object will be instantiated and called as such:  
// Codec codec = new Codec();  
// codec.decode(codec.encode(strs));
```

Kotlin

```
class Codec {  
    // Encodes a list of strings to a single string.  
    fun encode(strs: List<String>): String {  
    }  
  
    // Decodes a single string to a list of strings.  
    fun decode(s: String): List<String> {  
    }  
}  
  
/**  
 * Your Codec object will be instantiated and called as such:  
 * var obj = Codec()
```

```
* val s = obj.encode(strs)
* val ans = obj.decode(s)
*/
```

Go

```
type Codec struct {

}

// Encodes a list of strings to a single string.
func (codec *Codec) Encode(strs []string) string {
}

// Decodes a single string to a list of strings.
func (codec *Codec) Decode(strs string) []string {
}

// Your Codec object will be instantiated and called as such:
// var codec Codec
// codec.Decode(codec.Encode(strs));
```
