| 📄 Instructions | </> Code Examples | ▶ Screencasts | 🎓 Concept | **Stage #LA7** | **Pending** |

```
$ redis-cli SET foo bar
```

The tester will expect to receive `+OK\r\n` as a response (that's the string `OK` encoded as a **RESP simple string**).
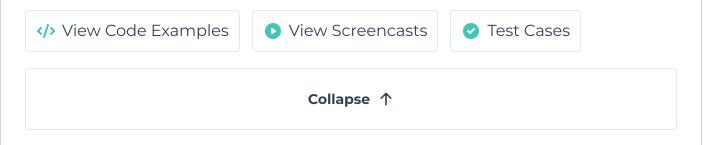
This command will be followed by a `GET` command:

```
$ redis-cli GET foo
```

The tester will expect to receive `$3\r\nbar\r\n` as a response (that's the string `bar` encoded as a **RESP bulk string**.

Notes

- If you implemented a proper Redis protocol parser in the previous stage, you should be able to reuse it in this stage.

- Just like the previous stage, the values used for keys and values will be random, so you won't be able to hardcode the response to pass this stage.

- If a key doesn't exist, the `GET` command should return a "null bulk string" (`$-1\r\n`). We won't explicitly test this in this stage, but you'll need it for the next stage (expiry).

| </> View Code Examples | ▶ View Screencasts | ✅ Test Cases |

Collapse ↑

Hints

Filter by R

Ready to run tests...   Show logs