and started transaction 2 from another connection:

```
$ redis-cli
> MULTI
OK
> INCR foo
QUEUED
```

If you then run `EXEC` in transaction 1, you should see the following:

```
> EXEC
1) OK
2) (integer) 42
```

`OK` is the response to `SET foo 41`, and `42` is the response to `INCR foo`.

And for transaction 2, running `EXEC` should return:

```
> EXEC
1) (integer) 43
```

43 is the response to `INCR foo`. The key `foo` was updated to `42` by transaction 1, and `INCR foo` further increments it to `43`.

Tests

The tester will execute your program like this:

```
$ ./spawn_redis_server.sh
```

The tester will then connect to your server as multiple Redis clients, and send multiple commands from each connection:

```
$ redis-cli MULTI
```