

### Question 1)

۱- دستور PING را اجرا کرده و خروجی آن را نمایش دهید.

First, the Redis server should be setup. Next, Redis client is run to write in the queries.

```
127.0.0.1:6379> ping
PONG
127.0.0.1:6379>
```

### Question 2)

۲- فرض کنید قرارداد کرده‌ایم برای ذخیره‌سازی کلمه عبور کاربرها از کلیدهایی به صورت user:USERNAME:pass استفاده کنیم. مثلاً کلید user:ali:pass حاوی کلمه عبور علی است. حال با استفاده از دستور SET کلمه عبور alice را برابر مقدار @li3 ست کنید.

```
127.0.0.1:6379> set user:alice:pass @li3
OK
127.0.0.1:6379>
```

### Question 3)

۳- با استفاده از دستور GET مقدار کلمه عبور alice را نمایش دهید.

```
127.0.0.1:6379> get user:alice:pass
"@li3"
127.0.0.1:6379>
```

#### Question 4)

۴- با استفاده از دستور MSET مقدار کلمه عبورهای سه کاربر دلخواه را ذخیره کرده و با استفاده از MGET مقدار آنها را نمایش دهید.

```
127.0.0.1:6379> mset user:Tina:pass 111 user:Sherlock:pass 222 user:Watson:pass 333
OK
127.0.0.1:6379> mget user:Tina:pass user:Sherlock:pass user:Watson:pass
1) "111"
2) "222"
3) "333"
127.0.0.1:6379>
```

#### Question 5)

۵- با استفاده از دستور EXISTS وجود یک کلید را بررسی کنید. پس از اجرای دستور del رو آن کلید، مجدداً خروجی EXISTS را نمایش دهید.

As can be seen, after using the “del” command, the key and its value would be deleted:

```
127.0.0.1:6379> exists user:Tina:pass
(integer) 1
127.0.0.1:6379> del user:Tina:pass
(integer) 1
127.0.0.1:6379> exists user:Tina:pass
(integer) 0
127.0.0.1:6379>
```

#### Question 6)

۶- فرض کنید در پایگاه داده‌ی یک سایت فروش کالا، ویش لیست هر کاربر را به صورت لیستی از نام کالاها تحت کلیدی به صورت USERNAME:wishlist نگهداری می‌کنیم. با استفاده از دستور RPush کالاهای pen، pencil، sharpner را در ویش لیست eric ذخیره کنید. سپس ویش لیست او را نمایش دهید.

```
127.0.0.1:6379> rpush eric:wishlist pen pencil sharpner
(integer) 3
127.0.0.1:6379> lrange eric:wishlist 0 -1
1) "pen"
2) "pencil"
3) "sharpner"
127.0.0.1:6379>
```

### Question 7)

۷- با استفاده از دستور RPOPLPUSH آخرین کالایی که در ویش لیست eric است را به لیست کالاهایی که او دیده است (USERNAME:visited) اضافه کنید.

```
127.0.0.1:6379> rpoplpush eric:wishlist eric:visited
"sharpner"
127.0.0.1:6379> lrange eric:visited 0 -1
1) "sharpner"
127.0.0.1:6379>
```

### Question 8)

۸- با استفاده از دستور SADD آدرس تعدادی سایت خبری را در کلید news و تعدادی سایت ورزشی را در کلید sport ذخیره کنید. محتویات هر دو کلید را نمایش دهید.

```
127.0.0.1:6379> sadd news "bbc.co.uk" "edition.cnn.com" "yahoo.com"
(integer) 3
127.0.0.1:6379> sadd sport "espn.com" "marca.com"
(integer) 2
127.0.0.1:6379> smembers news
1) "bbc.co.uk"
2) "yahoo.com"
3) "edition.cnn.com"
127.0.0.1:6379> smembers sport
1) "espn.com"
2) "marca.com"
127.0.0.1:6379>
```

### Question 9)

۹- مقدار کلید ice را برابر I'm melting قرار دهید به طوری که این کلید تنها برای ۱۰ ثانیه وجود داشته باشد و پس از آن به صورت خودکار حذف شود. حذف شدن آن را نمایش دهید. چگونه می‌توانیم پس از گذشت چند ثانیه مدت زنده ماندن یک کلید را تمدید کنیم؟

The key "ice" was set to last for ten seconds, and then before the 10 seconds was up, it was shown to have the value of "I'm melting":

```
127.0.0.1:6379> set ice "I'm melting" EX 10
OK
127.0.0.1:6379> get ice
"I'm melting"
127.0.0.1:6379>
```

After waiting for 10 seconds, the value for key “ice” will be automatically deleted as bellow:

```
127.0.0.1:6379> get ice
(nil)
127.0.0.1:6379>
```

For the last question of how we can extent the “TTL” (existence time remaining) of a key, two methods can be employed:

1. Using “persist <key>”: which extends the existence of the key to ever!
2. Using “expire <key> <seconds>”: which extends the existence of the key to the specified time limit

According to our preference, either of the methods can be used. Here are examples of them both in practice:

### Method 1) “persist”

First, key “hi” is created with an existence time limit of 20 seconds. As can be seen, the “ttl” command shows how much time is left until the key “hi” is deleted. Clearly, it’s decremented over time:

```
127.0.0.1:6379> set hi "Tina" EX 20
OK
127.0.0.1:6379> ttl hi
(integer) 17
127.0.0.1:6379> ttl hi
(integer) 13
```

Next, “persist” command is used to extend the key’s existence. As can be seen bellow, it will continue to exist forever since the “ttl” command is showing “-1”. (if the “ttl” command was showing “-2”, then that would mean that the key is deleted from the database!)

```
127.0.0.1:6379> persist hi
(integer) 1
127.0.0.1:6379> ttl hi
(integer) -1
127.0.0.1:6379> get hi
"Tina"
127.0.0.1:6379>
```

## Method 2) “expire”

First, key “hi” is deleted to be created again.

```
127.0.0.1:6379> del hi
(integer) 1
127.0.0.1:6379> get hi
(nil)
127.0.0.1:6379>
```

Next, just like in method 1, key “hi” is created again with an existence time limit of 20 seconds. As can be seen, the “ttl” command shows how much time is left until the key “hi” is deleted. Clearly, it’s decremented over time:

```
127.0.0.1:6379> set hi "George" EX 20
OK
127.0.0.1:6379> ttl hi
(integer) 17
127.0.0.1:6379> ttl hi
(integer) 12
```

Next, the “expire” command is used to extend the key’s existence time to 70 seconds instead. Here, as presented, it was successfully executed:

```
127.0.0.1:6379> expire hi 70
(integer) 1
127.0.0.1:6379> ttl hi
(integer) 67
127.0.0.1:6379> ttl hi
(integer) 63
127.0.0.1:6379> get hi
"George"
127.0.0.1:6379> ttl hi
(integer) 12
```

After the 70 seconds is up, the key will be automatically deleted: (i.e., the “ttl” command will show -2)

```
127.0.0.1:6379> ttl hi
(integer) -2
127.0.0.1:6379>
```

### Question 10)

۱۰- فرض کنید می‌خواهیم اطلاعات کاربرها را با کلید user:USERNAME در داده ساختار hash نگهداری کنیم. برای هر کاربر نام، نام‌خانوادگی، تاریخ تولد و کد ملی اش را در اختیار داریم. با استفاده از دستور hmset کاربری با نام کاربری john با داده‌های دلخواه وارد کنید. سپس ابتدا کلیه اطلاعات ذخیره شده برای وی را نمایش دهید و در ادامه تنها نام او را نمایش دهید.

First, all the related information for user “John” in the hash table is presented:

```
127.0.0.1:6379> hmset user:john name "John" familyname "Watson" birthday "1852-08-07" id 12345
OK
127.0.0.1:6379> hgetall user:john
1) "name"
2) "John"
3) "familyname"
4) "Watson"
5) "birthday"
6) "1852-08-07"
7) "id"
8) "12345"
127.0.0.1:6379>
```

Next, only John’s name is presented:

```
127.0.0.1:6379> hget user:john name
"John"
127.0.0.1:6379>
```