1. Sorting

Write a `sort` function asks the user to enter three numbers, and prints them out in an ascending order.

Ex:

```
>>> sort()
x = 3
y = 2
z = 1
Ordered: 1, 2, 3

>>> sort()
x = 4
y = 3
z = 4
Ordered: 3, 4, 4
```

---

2. Comparison

Write a `compare` function takes two values, `x` and `y`, and returns `1` if `x > y`, `0` if `x == y`, and `-1` if `x < y`.

Ex:

```
>>> compare(6,4)
1
```

---

3. Palindrome

Exercise 6.3

---

4. Power

Exercise 6.4

---

5. GCD

Exercise 6.5

---

6. Square Root

Exercise 7.1

---

7. Evaluation

Exercise 7.2

---

8. PIN Code

a) Write a mini ATM program that keeps prompting the user to enter the correct PIN code number. (`pin = 1234`; use `while`)

Ex:

```
>>> Enter your PIN number: 1111
Error! Enter again: 1222
Error! Enter again: 1224
Error! Enter again: 1225
Error! Enter again: 1234
Correct!
```

b) Modify your code so that the program stops if incorrect number has been entered three times in row.

Ex:

```
>>> Enter your PIN number: 1111
Error! Enter again: 1222
Error! Enter again: 1224
Error! Your card has been blocked.
```

----

9. Mini Calculator

Write a program that prompts the user to enter a number repeatedly , and prints out the current total sum each time, until 0 is entered.

Ex:

```
>>> Enter a number: 60
Current sum: 60
>>> Enter a number: 100
Current sum: 160
>>> Enter a number: 180
Current sum: 340
>>> Enter a number: 0
TOTAL: 340
```

----

10. Prime Numbers

Write a program that finds and prints out all the *prime numbers* from 1 to 1000.

Ex:

```
>>> prime()
1, 2, 3, 5, 7, 11, 13, …
```
