

## Python Exercises

**Exercise 1:** Use Python's help facility to learn how to

- remove an element from a list
- insert an element into a list
- sort a list

**Exercise 2:** Create a function `gcd(a, b)` that implements the Euclidean algorithm. The essential fact is that if  $a = bq + r$ , then  $\text{gcd}(a, b) = \text{gcd}(b, r)$ . The Euclidean algorithm repeatedly applies this observation until the remainder  $r$  is zero.

**Exercise 3:** Create a function `primes(n)` that returns a list of all primes less than or equal to  $n$ .

**Exercise 4:** Create a function `primefactorization(n)` that returns a dictionary whose keys are primes and whose values are the number of times that prime occurs in the prime factorization of  $n$ . For instance, `primefactorization(112)` returns the dictionary

```
{2:4, 7:1}
```

Use this function to write out the prime factorization of  $n$ . For instance, if  $n$  is 112, write out

```
2**4 7**1
```

**Exercise 5:** Create a function `bisection(f, a, b, tolerance)` that implements the bisection method to find a zero of the continuous function  $f$  to within `tolerance`. Remember that the bisection method works when  $f(a)$  and  $f(b)$  have opposite signs so that we know there is a zero in the interval  $[a, b]$ . We then narrow the interval containing the zero by half by considering the signs of  $f(a)$ ,  $f(\frac{a+b}{2})$ , and  $f(b)$ .

**Exercise 6:** Create a class `Vector` for two-dimensional vectors. Include functions that find the length of a vector, that normalize the vector, that find the dot product of a vector with another, and that find the angle between two vectors.