第十一周

1. Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (n) (both included) and the values are square of keys. The function should just print the values only.
   Hints:
   - Use `dict[key] = value` pattern to put entry into a dictionary.
   - Use `**` operator to get power of a number.
   - Use `range()` for loops.
   - Use `keys()` to iterate keys in the dictionary. Also we can use `item()` to get key/value pairs.

2. Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (n) (both included).
   Hints:
   - Use `**` operator to get power of a number.
   - Use `range()` for loops.
   - Use `list.append()` to add values into a list.

3. Define a function which can generate a list where the values are square of numbers between 1 and 20 (n) (both included). Then the function needs to print the first 5 elements in the list.
   Hints:
   - Use `**` operator to get power of a number.
   - Use `range()` for loops.
   - Use `list.append()` to add values into a list.
   - Use `[n1: n2]` to slice a list.

4. Define a function which can generate a list where the values are square of numbers between 1 and 20 (n) (both included). Then the function needs to print the last 5 elements in the list.

5. Define a function which can generate a list where the values are square of numbers between 1 and 20 (n) (both included). Then the function needs to print all values except the first 5 elements in the list.

6. It's a generally accepted belief, to assume that one year in the life a dog corresponds to seven years in the life of a human being. But apparently there are other subtle methods to calculate this haunting problem, haunting at least for some dog owners.
   Another subtler method works like this:
   A one-year-old dog roughly corresponds to a fourteen-year-old child.
   A dog who is two years old corresponds to a 22-year-old human.
   Every further dog year corresponds to five human years.
   Write a program to implement the dog-human-age rule.

7. Sieve of Eratosthenes:
   In mathematics, the sieve of Eratosthenes is an ancient algorithm for finding all prime numbers up to any given limit.
   It does so by iteratively marking as composite (i.e., not prime) the multiples of each prime, starting with the first prime number, 2. The multiples of a given prime are generated as a sequence of numbers starting from that prime, with constant difference between them that is equal to that prime. This is the sieve's key distinction from using trial division to sequentially test each candidate number for divisibility by each prime. Once all the multiples of each discovered prime have been marked as composites, the remaining unmarked numbers are primes.
   View more: https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes.

   - Write a function to determine whether a given integer is prime or not.
   - Write a function to find all the primes less than 1,000,000