**Exercise 1: Playing with Files and Strings.**

A KeWord In Context (KWIC) index is a simple index for a list of lines or titles. This assignment involves creating a KWIC index for an input list of titles stored in a file. Here's a small example. For the input file:

Casablanca

The Maltese Falcon

The Big Sleep

your program should produce the output:

3                              The Big Sleep    .

1                                  Casablanca   .

2                      The Maltese Falcon       .

2                              The Maltese Falcon

3                          The Big Sleep        .

As you can see, each title is listed for each word (omitting some **minor** words). The titles are arranged so that the word being indexed is shown in a column on the page. The position the lines have in the input file is shown on the left in the result.

Your Python solution should follow the following rules:

* The input is just a series of titles, one per line. Any leading or trailing spaces should be removed. Internal spaces should be retained (trimmed to one).
* A word is a maximal sequence of non-blank characters.
* The output line is at most 79 characters wide.
  + The number is 5 characters wide, right-justified.
  + There is a space after the number.
  + The key word starts at position 40 (numbering from 1).
  + If the part of the title left of the keyword is longer than 33, trim it (on the left) to 33.
  + If the part of the keyword and the part to the right is longer than 40, trim it to 40.
* Each title appears in the output once for each word that isn't **minor.** Any word of length two or less is minor, and and the words are minor words.
* If a title has a repeated word, it should be listed for each repetition.
* Sorting should be case-insensitive.

**Exercise 2: A Few More Math.**

Goldbach's conjecture is one of the oldest unsolved problems in number theory and in all of mathematics. It states:

Every even integer greater than 2 is a Goldbach number, i.e., a number that can be expressed as the sum of two primes.

Expressing a given even number as a sum of two primes is called a Goldbach partition of the number. For example,

04 = 2 +  2           6 = 3 +  3           8 = 3 +  5

10 = 7 +  3          12 = 5 +  7          14 = 3 + 11

16 = 5 + 11          18 = 7 + 11          20 = 7 + 13

Write the following Python functions:

* goldbach(n) that returns a Goldbach partition for n
* goldbach\_list(n,m) that returns a dictionary indexed on the even numbers in the range (n,m) and whose values are their Goldbach partition

**Exercise 3: A Matrix Class.**

You are to write a class Matrix for representing and manipulating integer matrices. An instance of this class stores the elements of a matrix as a list of lists of integers, and provides methods for the operations of matrix equivalence, matrix copy, matrix addition, scalar-matrix multiplication, matrix-matrix multiplication, matrix transposition, and matrix norm -- the "size" of a matrix. Override the appropriate operators and raise the appropriate exceptions.

We first define these operations, and then give a skeleton of the Matrix class showing the signatures for all methods and constructors you must implement.

**The operations**

Let aij denote the i,j-th element of matrix A, located at row i, column j. Using this notation, the matrix operations listed above may be defined precisely as follows:

* **matrix equivalence** A ≃ B where A in Zm×n, B in Zp×q when m = p and n = q and bij = aij for i = 1,...,m j = 1,...,n;
* **matrix copy** B = A where A, B in Zm×n (bij = aij for i = 1,...,m j = 1,...,n);
* **matrix addition** C = A + B where A, B, C in Zm×n (cij = aij + bij for i = 1,...,m j = 1,...,n);
* **scalar-matrix multiplication** B = aA where A, B in Zm×n, a in Z (bij = a·aij for i = 1,...,m j = 1,...,n);
* **matrix-matrix multiplication** C = A·B where A in Zm×p, B in Zp×n, C in Zm×n (cij = Σk=1, ..., p aik·bkj for i = 1,...,m j = 1,...,n);
* **matrix transposition** B = AT where A in Zm×n, B in Zn×m (bji = aij for i = 1,...,m j = 1,...,n);
* **matrix norm (matrix 1-norm)** a = ‖A‖ where a in Z, A in Zm×n (a = maxj Σi | aij | for i = 1,...,m j = 1,...,n).

Note that in each case we state the proper matrix dimensions for the operation to be valid. For example, when multiplying two matrices A and B, the number of columns of A must match the number of rows of B.