Assignment number 3
Arithmetic coding

Deadline: May, 3

We assume that a source of information is represented in `Python` as in assignment number 2: as a list `[("a`$_1$`",`w_1`),...,("a`$_n$`",`w_n`)]` of letters `a`$_i$ and integers w_i that count the number of times each letter `a`$_i$ appears in a typical string of data produced by the source.

The objective of this assignment is to implement arithmetic coding. You have to program two functions: `arithmetic_encode`, and `arithmetic_decode` that perform the corresponding tasks. More specifically:

- `arithmetic_encode(str,src,k)` takes as input a string `str` $\in \Sigma^n$, a source of information `src`, whose alphabet must be the set Σ of letters that appear in the input string, and a number `k` that indicates the precision of the integers α, β that give the endpoints of the intervals in the algorithm, so that the initial interval is $\alpha = 0$ and $\beta = 2^k - 1$.

  Its output is a binary codeword `bin` of `0` and `1` which is the arithmetic encoding of `str`.

- `arithmetic_decode(bin,src,k,len)` takes as input a binary string `bin` of binary digits zero and one, a source of information `src`, the number `k` that gives the precision of the integers α, β and γ, and an integer `len` containing the number n of letters of the original string whose code is `bin`, so that one knows when to stop decoding.

  Its output is a string `str` of letters of the alphabet of `src` which is the arithmetic decoding of `bin`.

Of course, the best test of correctness of the implementations is that after decoding the binary data output by the encoder one recovers the original string.

Also, you may check that the length in bits of the code `bin` of a string `str` agrees with the letter probabilities, as explained in course lectures.