Assignment number 4
Dictionary coding

Deadline: May, 17

The objective of this assignment is to implement dictionary coding. You have to program functions for encoding and decoding with methods LZ77, LZSS, LZ78 and LZW. Notice that the similarity between the first two and the last two makes very easy to implement LZSS and LZW, and even other variants, once the basic LZ77 and LZ78 have already been implemented.

The names of the functions must be: `LZ77_encode`, `LZ77_decode`, and analogously for the others, as in the description below.

The functions take care only of the first part of the encoding: the computation of tokens, and so the output will be a list of tokens. Transforming that list into a binary sequence can be done as explained in the theory lesson, but is not included as part of this assignment.

The data to be compressed is always a string `str` of characters of some alphabet (binary, latin, or any other alphabet of ASCII symbols). In the LZW algorithms the alphabet must be given as a parameter. The corresponding encoding is a list `tok` of tokens, each having the following structure, depending on the coding method:

- 3-tuples $(\theta, \lambda, \mathtt{a})$ with the first two components being integers (offset and length) and the last one a character (a string of length one), for LZ77;
- 2-tuples $(\mathtt{0}, \mathtt{a})$ with a flag bit `0` and a character, and 3-tuples $(\mathtt{1}, \theta, \lambda)$ with a flag bit `1` and two integers, for LZSS;
- 2-tuples $(\iota, \mathtt{a})$ with an integer and a character, for LZ78;
- just integers ι for LZW.

Functions to be implemented. The functions to be implemented, their input parameters and their outputs are:

- `LZ77_encode(txt,s,t)`. Parameters s and t are integers giving the size of the search buffer and the lookahead buffer.
  Outputs `tok`, a list of tokens of type $(\theta, \lambda, \mathtt{a})$ with $\theta \leq s$ and $\lambda \leq t$.

- `LZ77_decode(tok)` outputs `txt`.

- `LZSS_encode(txt,s,t,m)`. The first parameters as before, and `m` the smallest match length λ required for encoding the match using a token of the type $(1, \theta, \lambda)$. Typical values of m may be 3 or 4.
  Outputs `tok`, a list of tokens of type $(\mathtt{0}, \mathtt{a})$ or of type $(\mathtt{1}, \theta, \lambda)$ with $\theta \leq s$ and $m \leq \lambda \leq t$.

- `LZSS_decode(tok)` outputs `txt`.

- `LZ78_encode(txt)`. At the beginning the dictionary contains only the empty word and there is no bound for the dictionary size.
  Outputs `tok`, a list of tokens of type $(\iota, \mathtt{a})$.

- `LZ78_decode(tok)` outputs `txt`.

- `LZW_encode(txt,alp)`. The parameter `alp` is a list containing the alphabet of letters in `txt`. At the beginning the dictionary contains all the one-letter words: the letters in `alp`, in the same given order, and there is no bound for the alphabet size.
  Outputs `tok`, a list of integers pointing to dictionary words.

- `LZW_decode(tok,alp)` outputs `txt`. The parameter `alp` must be the same passed to the encoding function.