**Problem A. Discrete random variables**

Write a class called Drv (discrete random variable) for the interpretation of finite discrete random variables. The constructor of the class waits for two lists of the same length, the first contains the values *xk* of the random variable *X* sorted increasingly, the second contains the probabilities *pk=P(X=xk)*.

Write the following methods:

* \_\_init\_\_: constructor defines two attributes *xk* and *pk* for the two lists.
* pdf(x): probability density function of *X*, returns *pk* if *x=xk*, otherwise return 0.
* cdf(x): cumulative distribution function of *X*, returns *P(X<xk).*
* e(): returns *E(X)*, the expected value of *X*.
* is\_nonneg(): this method returns True if the random variable is non-negative, otherwise returns False.
* reweight(): this method returns an instance of a random variable *Y* from the original *X* with the next "reweighting": *P(Y=xk*)=*xk\*pk/E(X*), if *X* is non-negative and has at least one nonzero value.
* Binomial: A derived class from Drv for the binomial distribution. Parameters of this class are *n* and *p*. Rewrite the code in this class for the e and for the is\_nonneg methods.
* Uniform: A derived class from Drv for the uniform distribution. The parameter of this class is *n*, the list of values is *[1,2,...,n]*. Rewrite the methods mentioned above*.*

Here you find some tutorials about: az [classes](https://www.w3schools.com/python/python_classes.asp) and a [derived classes](https://www.w3schools.com/python/python_inheritance.asp). The task is to complete the attached python code (4A.py) such that the classes work as described above.