Algorithm 5.1 The k-fold cross-validation algorithm. It can be used to estimate generalization error of a learning algorithm A when the given dataset $\mathbb{D}$ is too small for a simple train/test or train/valid split to yield accurate estimation of generalization error, because the mean of a loss L on a small test set may have too high variance. The dataset $\mathbb{D}$ contains as elements the abstract examples $\boldsymbol{z}^{(i)}$ (for the i-th example), which could stand for an (input,target) pair $\boldsymbol{z}^{(i)} = (\boldsymbol{x}^{(i)}, y^{(i)})$ in the case of supervised learning, or for just an input $\boldsymbol{z}^{(i)} = \boldsymbol{x}^{(i)}$ in the case of unsupervised learning. The algorithm returns the vector of errors $\boldsymbol{e}$ for each example in $\mathbb{D}$, whose mean is the estimated generalization error. The errors on individual examples can be used to compute a confidence interval around the mean (equation 5.47). While these confidence intervals are not well-justified after the use of cross-validation, it is still common practice to use them to declare that algorithm A is better than algorithm B only if the confidence interval of the error of algorithm A lies below and does not intersect the confidence interval of algorithm B.

Define `KFoldXV`($\mathbb{D}, A, L, k$):
Require: $\mathbb{D}$, the given dataset, with elements $\boldsymbol{z}^{(i)}$
Require: A, the learning algorithm, seen as a function that takes a dataset as input and outputs a learned function
Require: L, the loss function, seen as a function from a learned function f and an example $\boldsymbol{z}^{(i)} \in \mathbb{D}$ to a scalar $\in \mathbb{R}$
Require: k, the number of folds

  Split $\mathbb{D}$ into k mutually exclusive subsets $\mathbb{D}_i$, whose union is $\mathbb{D}$.
  for i from 1 to k **do**
    $f_i = A(\mathbb{D} \backslash \mathbb{D}_i)$
    for $\boldsymbol{z}^{(j)}$ in $\mathbb{D}_i$ **do**
      $e_j = L(f_i, \boldsymbol{z}^{(j)})$
    end for
  end for
  Return $\boldsymbol{e}$