task 3

the following article explains an algorithm that solves our problem: https://en.wikipedia.org/wiki/Reservoir_sampling#Simple:_Algorithm_R

the main idea is to keep a reservoir of k items, deciding randomly if the i-th item you revealed should be added to the reservoir (replacing any item already in there at that slot), otherwise discarding the item.

the random variable determining the inclusion of the item in the reservoir is given as a random integer between 1 and i; if it is less than or equal to k, it will fit in the reservoir and then occupies that drawn slot in the reservoir. if it is greater than k, it will be discarded.

this ensures a sampling with a uniform probability distribution of n

proof: since an item i is included in the reservoir with a probability of $\frac{k}{i}$, the next item $i+1$ has probability $\frac{k}{i+1}$ of being included in the reservoir. but since processing the $(i+1)$-th input affects the probability of the previous elements, the probability of i is now $\frac{k}{i} \times \left(1 - \frac{1}{i+1}\right) = \frac{k}{i+1}$. but then it isn't actually the case that the probability is different for the previous elements, i.e. the sampling was uniform and independent.

as the wiki mentions, the algorithm is needlessly slow, but plenty apt for a simple solution to our problem.