1. **Reservoir sampling** is a method for choosing an item uniformly at random from an arbitrarily long stream of data; for example, the sequence of packets that pass through a router, or the sequence of IP addresses that access a given web page. Like all data stream algorithms, this algorithm must process each item in the stream quickly, using very little memory.

```plaintext
GETONESAMPLE(stream S):
    ℓ ← 0
    while S is not done
        x ← next item in S
        ℓ ← ℓ + 1
        if RANDOM(ℓ) = 1
            sample ← x  (★)
    return sample
```

At the end of the algorithm, the variable \( ℓ \) stores the length of the input stream \( S \); this number is not known to the algorithm in advance. If \( S \) is empty, the output of the algorithm is (correctly!) undefined.

Consider an arbitrary non-empty input stream \( S \), and let \( n \) denote the (unknown) length of \( S \).

(a) Prove that the item returned by \( \text{GETONESAMPLE}(S) \) is chosen uniformly at random from \( S \).

(b) What is the exact expected number of times that \( \text{GETONESAMPLE}(S) \) executes line (★)?

(c) What is the exact expected value of \( ℓ \) when \( \text{GETONESAMPLE}(S) \) executes line (★) for the last time?

(d) What is the exact expected value of \( ℓ \) when either \( \text{GETONESAMPLE}(S) \) executes line (★) for the second time (or the algorithm ends, whichever happens first)?

(e) Describe and analyze an algorithm that returns a subset of \( k \) distinct items chosen uniformly at random from a data stream of length at least \( k \). The integer \( k \) is given as part of the input to your algorithm. Prove that your algorithm is correct.

For example, if \( k = 2 \) and the stream contains the sequence \( ⟨ ♠, ♥, ♦, ♣ ⟩ \), the algorithm would return the subset \( \{ ♠, ♣ \} \) with probability \( 1/6 \).
(a) Prove that the item returned by `GETONESAMPLE(S)` is chosen uniformly at random from $S$.

(b) What is the exact expected number of times that `GETONESAMPLE(S)` executes line ($\star$)?

(c) What is the exact expected value of $\ell$ when `GETONESAMPLE(S)` executes line ($\star$) for the last time?

(d) What is the exact expected value of $\ell$ when either `GETONESAMPLE(S)` executes line ($\star$) for the second time (or the algorithm ends, whichever happens first)?

(e) Describe and analyze an algorithm that returns a subset of $k$ distinct items chosen uniformly at random from a data stream of length at least $k$. The integer $k$ is given as part of the input to your algorithm. Prove that your algorithm is correct.