(Q1.) Theory

We have an initial capital of 1$ and the flip of a fair coin (P(H)=1/2 and P(T) = 1/2)) decides whether there is a profit or a loss. **If we obtain a Heads, a profit of 1$ is obtained**. **On Tails, we incur a loss of 0.5$**.

The assumption is made that once the current capital reaches 0$, no further flips will take place, and the trial will end. The reason for this assumption is the use of the words profit and loss explicitly in the question language, implying a system of balance and modifications to that balance, akin to gambling but with fixed returns.

The simulate function fulfills the role of simulating the whole trial of N flips and stops either when an N number of flips have already been done or if the balance reaches 0$. To simulate the actual flip, the python random module is used for its choice function, which takes an iterable and returns a random element from the iterable with all elements having the same chance of being selected. This is used on the sample space of Heads or Tails, and the total is modified accordingly.

The run function uses the multiprocessing module from python for its asynchronous programming capabilities to speed up the entire simulation when needed (at a higher number of passes or N) by dividing the total number of trials into several chunks and parallelly processing them.

The trials are, in total, conducted a passes number of times, and all the results (capital left at end of the trial) are added to a list. This list is then used in the transformToRV function to create a dictionary that stores the result as the key and the number of its occurrences in the entire simulation as the value. This creates an RV, say X, where X is the different amounts of final capital. Their respective probabilities are then calculated from the number of occurrences over the total values.

Since the entire simulation can be considered as returning a sample of the different values of the final capital and not the entire population, the mean and variance are calculated by the analyze function with passes-1 degrees of freedom.

In theory, as the number of passes tends to infinity, the sample mean and variance we calculate here should be equal to the actual population mean and variance. So, at a large number of passes, we can consider the obtained values to be approximately correct to at least a particular resolution while keeping in mind their approximate convergence (while varying the number of passes).

Note that these values may be calculated reasonably easily using probability mathematics, but the question has explicitly asked to use code for the same.

The obtained values are as follows:

N = 10; passes = 1000000; mean = 2.91 and variance = 6.99

N = 20; passes = 1000000; mean = 4.51 and variance = 17.29

These values may be a bit inaccurate (may have a significant deviation from actual values) since the number of passes is less than 2^N, and as a result, not all the trial outcomes (combinations of Heads and Tails) may be observed in the entire simulation, skewing the probabilities; Increasing the number of passes is not a solution since it would require an absurd amount of computation time.

N = 400; passes = 1000000; mean = 63.21 and variance = 2610.06

N = 10000; passes = 1000000; mean = 1555.5 and variance = 1475900.08

N = 100000; passes = 100000; mean = 15646.23 and variance = 146465170.47