**Tasks for Rayhan**

* (DONE) Output given complete patient paths, or better as simulation is run, for each year, the death % of the population i.e number of people in the “death” node. The years are bounded by number of steps each simulation step is one year
  + Do a small matplotlib viz
  + Test for correctness and time function on 10\*\*6 size state matrix, 50 steps. Report result
* (DONE) Do the input logic
  + Create an excel similar to “Inputs\_CEA.xlsx” reflecting all the inputs we need take in and outputting transition\_matrix, rewardVector, utilityVector in the excel.
  + Update the code to read these objects and pass them successfully through a simulation run using realLifeSimulation()
  + Any intermediate variables used in the excel, create functions to read them and create auxiliary functions for generating them (for ranges) for sensitivity purposes.
    - These variables should include HCC-early, HCC-late proportions for both control and intervention. The range-generator functions above will allow us to build the cost-effectiveness frontier below.
* Extend survival rate curves to each state transitioning to death and output these.
* Do the output logic cost-effectiveness frontier plot first draft
  + Build cost-effectiveness plot based on my cost-effectiveness logic (as a lambda function). Can output dollar difference, then binary function (cost-effective/not cost-effective)
  + For a fixed intervention scenario, produce one plot. Plot should have axes x being early-stage HCC prop in control and y being late stage HCC prop in control
  + Line in phase space should be for fixed parameters, represents cost-effectiveness frontier relative to fixed intervention scenario. Shade cost-effective region. Vary the HCC incidence rate in MASLD to obtain different cost-effectiveness lines in a separate plot.
  + Line in phase space will be jagged because of discretisation of phase-plane sensitivity parameters. Fit polynomial line to jagged frontier.
  + Produce the above plots for specific sub-populations.
  + Create logic to store plots in a given folder for each output run.
  + Report run times.