We first set up the same set of basic assumptions and variables.

```
GRAV <- 9.8 # gravity (m/s^2)
MASS <- 1 # mass (kg)
I_CM <- 1/12 # roational inertia (kg m^2)
PHI <- 0.1 # angle of Ft relative to floor (rad)
L <- 0.5 # distance from the center of mass (of rotation point) to tenson (m)
FT <- 11 # tension force (N)
OMEGA <- 0.1 # angle of floor relative to gravity (rad) (because shifted axis)
```

Additionally, we set the time interval and seed values for all values that's tallied:

```
dt <- 0.05
t_max <- 5

time <- 0
theta <- 0
thetadot <- 0
```

Great. Let's start writing the loop now by setting up a bunch of arrays and writing their values in.

```
cTime = NULL
cTorqueNet = NULL

cDDTheta = NULL
cDTheta = NULL
cTheta = NULL
cFNetX = NULL
cFNetY = NULL
```

Awesome, we will start tallying, then!

```
for (i in 0:(t_max/dt)) {
    # write down standard values
    cTime[i] = time
    cTheta[i] = theta

    # torque is calculated via the dot product between the vector of the radius projected out
    # and also the angle at which the thing is at (so like theta + phi)
    #
    # note that, unlike the tabled version, L here represets distance from CoM to tension
    # application
    torque <- FT*L*cos(theta+PHI)
    cTorqueNet[i] = torque

    # from knowing the torque, we could divide out the rotational inertia to figure the
    # acceleration of rotation
    thetadotdot <- torque/I_CM
    cDDTheta[i] <- thetadotdot



    # from this, we could of course tally for the velocity of theta as well
```

```
    thetadot <- dt*thetadotdot + thetadot
    cDTheta[i] <- thetadot

    # We increment the time and theta based on the tallying variable
    time <- dt + time
    theta <- dt*thetadot + theta
}
```