I. WIGGLES IN THE PRIMORDIAL POWER SPECTRUM

Modify CAMB to add some features in the primordial power spectrum in order to change power at large scales.
A simple parameterization, not connected to any particular physical model, could be

$$\mathcal{P}(k) = \left[1 + \frac{\Delta\mathcal{P}(k)}{\mathcal{P}_{std}(k)}\right] \mathcal{P}_{std}(k) \tag{1}$$

with

$$\frac{\Delta\mathcal{P}(k)}{\mathcal{P}_{std}(k)} = A\ e^{-\mu k/k_0} \sin(\nu k) \tag{2}$$

A, μ and ν are new parameters and need to be defined and read.
Since we are dealing with initial power spectrum parameters, everything can be contained in `power_tilt.f90`

- define new parameters in the `InitialPowerParams` type
- read them in the `InitialPower_ReadParams` routine
- set their default value in the `SetDefPowerParams` routine
- include them in the `params.ini`

Tip: follow an existing parameter within the code (e.g. `ScalarPowerAmp`) to understand what to do

Implement Eqs.(1-2) in the function where the primordial power spectrum is computed (`ScalarPower`).
Play with the parameter values and plot the C_ℓ^{TT}; try to lower the low-ℓ tail of the spectrum without spoiling the first peak!

II. ADD AN EXTRA PARAMETERIZATION TO EFTCAMB

EFTCAMB contains default parameterizations for the time evolution of the EFT functions.
It is also straightforward to implement your own favourite parameterization.

Add a new, JBP inspired, parameterization for $\Omega(a)$ in the code:

$$\Omega(a) = \Omega_0 + \Omega_a a(1-a) \tag{3}$$

You can add extra parameters to the code or use the already existing ones, e.g.

$$\Omega(a) = \mathrm{EFTOmega0} + \mathrm{EFTOmegaExp}(1-a)a \tag{4}$$

- include the new expression for $\Omega(a)$ in `EFT_functions.f90`
- include derivatives of the parameterization in `EFT_functions.f90`

Tip: use the already existing space for a user defined parameterization (remember to comment out the `stop`)

Once this is done, run the code (choosing the user defined parameterization) and compare with the results of the parameterizations used during the lectures.

What's the main physical difference with the default parameterizations? What is "wrong" with this choice given the standard behaviour of the code?