```python
from q2_c import *

# ==========================================
# Find T(t) from t_rh to today!
# ==========================================

logt_today = np.log(6.62e41)
T_today = T_approx_exp(a_piecewise(logt_today))

# ==========================================
# Find H0
# ==========================================

H0_GeV = (np.pi / (3 * np.sqrt(10))) * np.sqrt(g_star(T_today)) * (T_today ** 2) / M_pl  # GeV

print("H0 (GeV): {}".format(H0_GeV))

# ==========================================
# Find critical density for universe
# ==========================================

# Conversion Factor
inverse_gev_to_seconds = (1.52e24)

# Convert H0 to seconds
H0_seconds = (1 / H0_GeV) / inverse_gev_to_seconds
print("H0 (Seconds): {}".format(H0_seconds))

# Convert M_pl to si units
G = 6.67e-8  # grams/ cm3 s2
M_pl_sq = 1 / (8 * np.pi * G)

# Calculate critical density
critical_rho = 1 / ((3 * H0_seconds ** 2) / M_pl_sq)

print("Critical Density (g / cm3): {}".format(critical_rho))

# ==========================================
# Calculate Vacuum Density
# ==========================================

omega_vac = 0.68
rho_DM_gcm3 = omega_vac * critical_rho

print("Vacuum Density (g / cm3): {}".format(rho_DM_gcm3))

gcm3_to_ev4 = 2.32e17 * (1 / (1e9) ** 4)
rho_DM_ev4 = rho_DM_gcm3 / gcm3_to_ev4
print("Vacuum Density (eV^4): {}".format(rho_DM_ev4))

# ==========================================
# Calculate Cosmological Constant Today
# ==========================================

c = 1e8
CC = rho_DM_gcm3 * (8 * np.pi * G / c ** 2)
```

```
57 print("Cosmological Constant Today (1/m^2): {}".format(CC))
58
59 # ========================================
60 # Calculate Cosmological Constant before EW
61 # ========================================
62
63 # Same process as before but with different T0!!
64 T0_EW = 100  # GeV
65 H0_GeV_EW = (np.pi / (3 * np.sqrt(10))) * np.sqrt(g_star(T0_EW)) * (T0_EW ** 2
   ) / M_pl  # GeV
66 inverse_gev_to_seconds = (1.52e24)
67 H0_seconds = (1 / H0_GeV_EW) / inverse_gev_to_seconds
68 G = 6.67e-8  # grams/ cm3 s2
69 M_pl_sq = 1 / (8 * np.pi * G)
70 critical_rho = 1 / ((3 * H0_seconds ** 2) / M_pl_sq)
71 omega_vac = 0.68
72 rho_DM_gcm3 = omega_vac * critical_rho
73 c = 1e8
74 CC_EW = rho_DM_gcm3 * (8 * np.pi * G / c ** 2)
75
76 print("Ratio b/w CC's: {}".format(CC_EW / CC))
77
```