Shi Rui's Answer to Question 2-8.2

Please refer to the Python codes listed below

```
# Convert the rate constant to per second
k1 = 100.0/60.0  # /s
k2 = 600.0/60.0
k3 = 150.0/60.0

# Define the initial concentrations in μM
E0 = 1
S0 = 10
ES0 = 0
P0 = 0

# copied the four equations derived from A 8.1 here as functions
def dEdt(E, S, ES, P):
    return k2*ES - k1*E*S

def dSdt(E, S, ES, P):
    return k2*ES - k1*E*S + k3*ES

def dESdt(E, S, ES, P):
    return k1*E*S - (k2 + k3)*ES

def dPdt(E, S, ES, P):
    return k3*ES

# Define the initial values for the four concentrations
E = E0
S = S0
ES = ES0
P = P0
```

```
# set up lists for each item with respective initial values
listE = [E]
listS = [S]
listES = [ES]
listP = [P]

h = 0.01 # a random time step
i = 0      # used i as an index for the list for the computation
# fourth-order Runge-Kutta method
while i < 30:
    F1_E = h * dEdt(listE[i], listS[i], listES[i], listP[i])
    F2_E = h * dEdt(listE[i]+0.5*h*F1_E, listS[i]+0.5*h*F1_E, listES[i]+0.5*h*F1_E, listP[i]+0.5*h*F1_E)
    F3_E = h * dEdt(listE[i]+0.5*h*F2_E, listS[i]+0.5*h*F2_E, listES[i]+0.5*h*F2_E, listP[i]+0.5*h*F2_E)
    F4_E = h * dEdt(listE[i]+h*F3_E, listS[i]+h*F3_E, listES[i]+h*F3_E, listP[i]+h*F3_E)
    dEdtH = round(listE[i] + 1 / 6 * (F1_E + 2 * F2_E + 2 * F3_E + F4_E),3)
    listE.append(dEdtH)

    F1_S = h * dSdt(listE[i], listS[i], listES[i], listP[i])
    F2_S = h * dSdt(listE[i]+0.5*h*F1_S, listS[i]+0.5*h*F1_S, listES[i]+0.5*h*F1_S, listP[i]+0.5*h*F1_S)
    F3_S = h * dSdt(listE[i]+0.5*h*F2_S, listS[i]+0.5*h*F2_S, listES[i]+0.5*h*F2_S, listP[i]+0.5*h*F2_S)
    F4_S = h * dSdt(listE[i]+h*F3_S, listS[i]+h*F3_S, listES[i]+h*F3_S, listP[i]+h*F3_S)
    dSdtH = round(listS[i] + 1 / 6 * (F1_S + 2 * F2_S + 2 * F3_S + F4_S),3)
    listS.append(dSdtH)

    F1_ES = h * dESdt(listE[i], listS[i], listES[i], listP[i])
    F2_ES = h * dESdt(listE[i]+0.5*h*F1_ES, listS[i]+0.5*h*F1_ES, listES[i]+0.5*h*F1_ES,
listP[i]+0.5*h*F1_ES)
    F3_ES = h * dESdt(listE[i]+0.5*h*F2_ES, listS[i]+0.5*h*F2_ES, listES[i]+0.5*h*F2_ES,
listP[i]+0.5*h*F2_ES)
    F4_ES = h * dESdt(listE[i]+h*F3_ES, listS[i]+h*F3_ES, listES[i]+h*F3_ES, listP[i]+h*F3_ES)
    dESdtH = round(listES[i] + 1 / 6 * (F1_ES + 2 * F2_ES + 2 * F3_ES + F4_ES),3)
    listES.append(dESdtH)
```

```
    F1_P = h * dPdt(listE[i], listS[i], listES[i], listP[i])
    F2_P = h * dPdt(listE[i]+0.5*h*F1_P, listS[i]+0.5*h*F1_P, listES[i]+0.5*h*F1_P, listP[i]+0.5*h*F1_P)
    F3_P = h * dPdt(listE[i]+0.5*h*F2_P, listS[i]+0.5*h*F2_P, listES[i]+0.5*h*F2_P, listP[i]+0.5*h*F2_P)
    F4_P = h * dPdt(listE[i]+h*F3_P, listS[i]+h*F3_P, listES[i]+h*F3_P, listP[i]+h*F3_P)
    dPdtH = round(listP[i] + 1 / 6 * (F1_P + 2 * F2_P + 2 * F3_P + F4_P),3)
    listP.append(dPdtH)

    i = i+1

print(listS)
print(listP)
```