Assignment 3

1)

X	Y	Z	W	Output
ON	ON	ON	ON	ON
ON	ON	OFF	OFF	OFF
OFF	ON	OFF	ON	OFF
ON	ON	OFF	ON	ON

2)

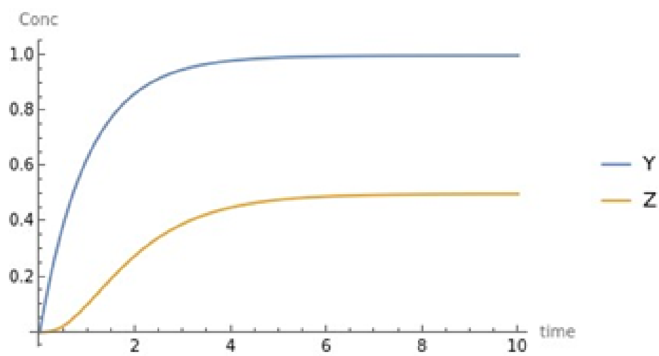
a)

To simulate the dynamics of C1-FFL we can use following lines of code:

```
# Question 2: Part a

def FFL(x , t , k = 1):

  beta , alpha , Sx = 1 , 1 , 1

  y = x[0]
  z = x[1]

  p = y / k

  dydt = beta * Sx - alpha * y
  dzdt = beta * Sx * (((p) ** 2) / (1 + p ** 2)) - alpha * z


  return [dydt , dzdt]

x0 = [0 , 0]

t = np.linspace(0,10)

x = odeint(FFL , x0 , t)

y = x[: , 0]
z = x[: , 1]

plt.figure(dpi = 150)
```

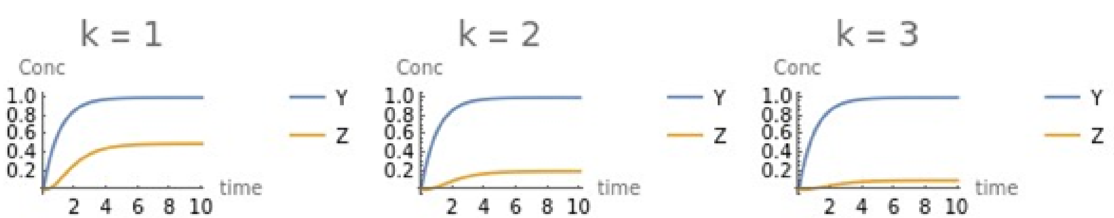
```
plt.plot(t,y / y[-1] , label = 'y[t]')
plt.plot(t,z / z[-1] , label = 'z[t]')
plt.xlabel('time')
plt.ylabel('conc[t]/steady state concentration')

plt.legend()

plt.title('Dynamics of FFL')

plt.show()
```

```
param = {β -> 1, Sx -> 1, α -> 1, n -> 2 , k -> 1}
soly = DSolve[{y'[t] == Sx * β - α * y[t] , y[0] == 0}/.param ,y , {t , 0 , 10}]
solz = NDSolve [{z '[t] == β * Sx * ((y[t] / k)^ n / (1 + (y[t] / k)^ n)) - α * z[t] , z[0] == 0}
/. param /. soly , z , {t , 0 , 10}]
Plot[{(y[t] /. soly), (z[t] /. solz)}, {t, 0, 10}, PlotRange → All, PlotLegends → {"Y", "Z"},
AxesLabel → {"time", "Conc"}]
```



b)

The delay in response time of gene z increases with the increase in k




```
# Question 2b: Changing values of K

plt.figure(dpi = 150)

t = np.linspace(0,8)
```

```
for k in range(1 , 10 , 5):

  z = odeint(FFL , x0 , t , args = (k , ))[: , 1]

  plt.plot(t , z/z[-1] , label = 'K = ' + str(k))

plt.legend()

plt.xlabel('time')
plt.ylabel('conc[t]/steady state concentration')

plt.title('Effect of K on response time')

plt.show()
```

c)

With positive regulation on Y, the model equations will be:

$$y'[t] = S_x * \beta * \frac{(y[t]/k_y)^{n1}}{1 + (y[t]/k_y)^{n1}} - \alpha * y[t]$$

$$z'[t] = S_x * \beta * \frac{(y[t]/k_{yz})^{n1}}{1 + (y[t]/k_{yz})^{n1}} - \alpha * z[t]$$

3)

a)

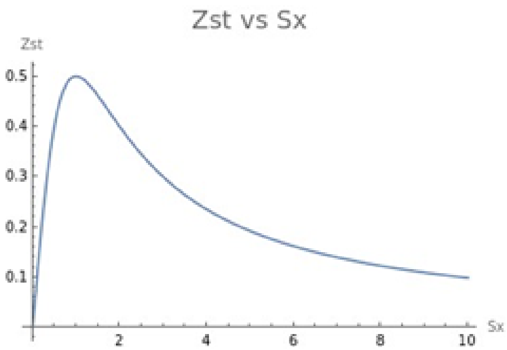
At steady state, $\frac{dy}{dt} = \frac{dz}{dt} = 0$

$$=> y'[t] = S_x - \beta * y_{st} = 0$$

$$=> y_{st} = \frac{S_x * \beta}{\alpha} = S_x (\beta = \alpha = 1)$$

$$z'[t] = \frac{S_x * \beta}{1 + (y_{st}/k)^2} - \alpha * z_{st} = 0$$

$$z_{st} = \frac{S_x}{1 + S_x^2}$$




```
# Question 3: Part a

# At steady state, dy / dt = dz / dt = 0

# y'[t] = Sx * beta - alpha * Yst = 0 (Yst = Steady state concentration of y)
# Yst = Sx * beta / alpha = Sx (alpha = beta = 1)

# z'[t] = Sx * beta / (1 + (Yst / k) ** 2) - alpha * Zst= 0
# Zst = Sx / (1 + Sx ^ 2)

Sx = np.linspace(0 , 10 , 100)

Zst = []

for ele in Sx:

  Zst.append(ele / (1 + ele ** 2))

plt.figure(dpi = 150)

plt.plot(Sx , Zst)

plt.xlabel("Sx")
plt.ylabel("Steady state concentration of Z")

plt.title("Zst vs Sx")
```

b)

To calculate activation signal Sx, at which Zst is maximum we need to find Sx at which $\frac{dz}{dS_x} = 0$

$$=> \frac{dz}{dS_x} = \frac{1 + S_x^2 - 2 * S_x^2}{(1 + S_x^2)^2} = 0$$

$$\frac{dz}{dS_x} = \frac{1 - S_x^2}{1 + S_x^2} = 0$$

$$=> S_x = 1$$