DATE: SUBJECT:

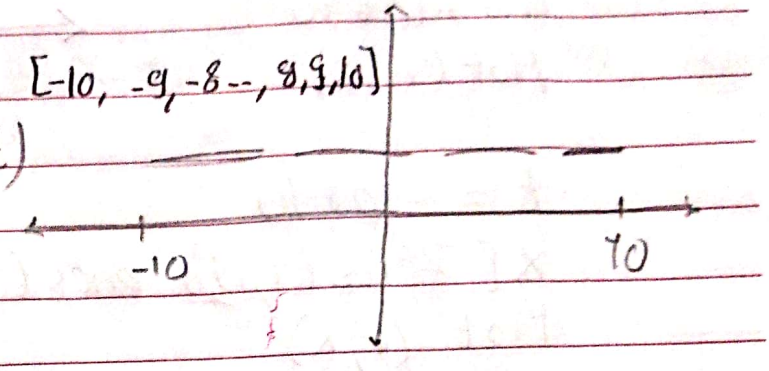
$$x(t) = \begin{cases} \cos(2\pi t) & -10 \le t \le 10 \\ 0 & \text{otherwise} \end{cases}$$

write a matlab code to plot the given signal.

(1×21 columns) ← t = -10 : 10 ⇒ [-10, -9, -8 --, 8, 9, 10]

```
x = cos(2*pi*t)
plot(t,x)
```

→ [||| ||| ||--]



This is wrong

we add a step

(201 columns) ← t = -10 : 0.1 : 10

The smaller the step, the more points we plot, the more memory, the more time in execution.

t = -10 : 0.01 : 10 (2001 columns)

$$x(t) = e^{-t/8} \qquad -10 \le t \le 10$$

```
t = -10: 0.1 : 10
X = exp(-t/8)
plot(t,X)
```

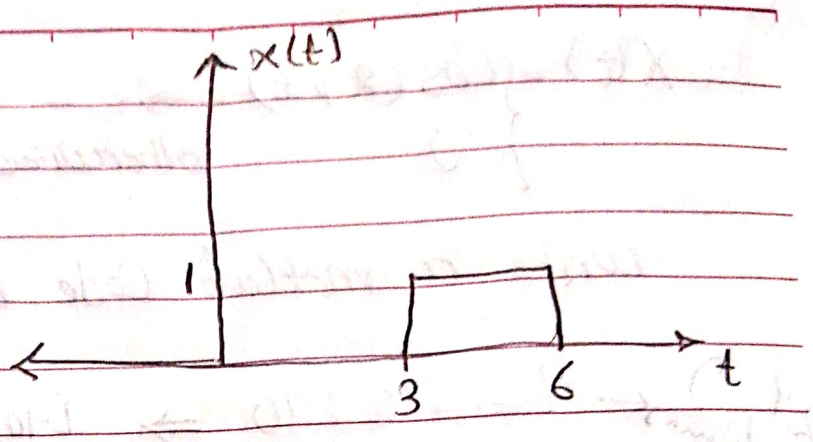
$$x = \cos(2\pi t)\, e^{-(t/8)} \qquad -10 \le t \le 10$$

```
t = -10: 0.1 : 10
X = cos(2*pi*t) .* exp(-t/8)
plot(t,X)
```

1×20 1×20

DATE: SUBJECT:

```
n = 3:6
x = 1
y = zeros( )
x = [y x y]
n = -10:10
plot(n,x)
```




```
t = -10:10
x[ zeros(1,4) ones(1,4) zeros(1,4) ]
Plot (t,x)
```

in Continous signals, avoid the step of one.

↓

~~t=-10:1:10~~

```
t = -10:0.1:-10
x = (t>=3) & (t<=6)
plot (t,x)
```

⇒ Just like the if( ) else { statement

use logical statement