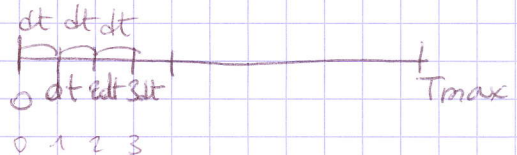


intervalle de temps  $[0, T_{\max}]$ :Q1: def init\_T( $T_{\max}, dt$ ):

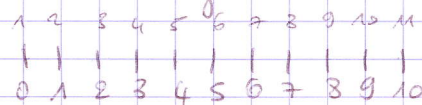
```

T = []
if  $T_{\max} \% dt == 0$ :
    for i in range(1,  $(T_{\max} // dt) + 1$ ):
        T.append(i * dt)
    return T
else:
    N =  $T_{\max} // dt + 1$ 
    T.append(i * dt)
    return T

```

ex: On divise  $[0, 10]$  en N

intervalles réguliers de taille 1



→ soit M éléments

$$\frac{10}{1} = 10$$

$$N = 10 + 1$$

$$\frac{T_{\max}}{dt} = N \text{ on veut } T_{\max} \text{ donc } N' = N + 1$$

Si le dernier élément n'est pas  $T_{\max}$  alors  $N' = N + 1$  donne le dernier élément et  $N'' = N + 2$  donne l'élément suivant.

Q2:  $e(t)$  sera représenté sous forme de liste  $E[i] = e(t_i)$ def init\_E( $T, f$ ): $N = \text{len}(T)$  $E_{\min} = 1,3$  $E_{\max} = 1,5$  $E = []$  $i = 0$ while  $i < N$  and  $T[i] < 16 \times \frac{1}{f}$ : $E.append(E_{\min} * \sin(2\pi f T[i]))$  $i = i + 1$ while  $i < N$  and  $T[i] < 2 \times 16 \times \frac{1}{f}$ : $E.append(E_{\max} * \sin(2\pi f T[i]))$  $i = i + 1$ while  $i < N$ : $E.append(\dots)$   $i = i + 1$ 

return(E)

def init\_E( $T, f$ ): $N = \text{len}(T)$  $E = []$ message =  $[0, 1, 0]$ 

for i in range(N):

 $i = i // (16/f)$ 

val = message[i]

if val == 0:

 $E_0 = 1,3$ 

else:

 $E_0 = 1,5$  $E.append(E_0 * \sin(2\pi f T[i]))$ 

return(E)

erreur ds corrigé.