

$$x(t) = \sin(2\pi f t) \quad (1)$$

$t : [0 ; 2]$ en [s]

$f = 1 \text{ Hz}$

l'espace entre 2 points deems : $0,001 \text{ [s]}$

① Question: combien de points pour représenter le signal x dans mon tableau?

#define N 2000

$$2s / 0,001 = 2000$$

② création du tableau

type : double

identification : x

→ Calculez et afficher la taille du tableau en octets.

2000 cases de type double = 16000 octets.

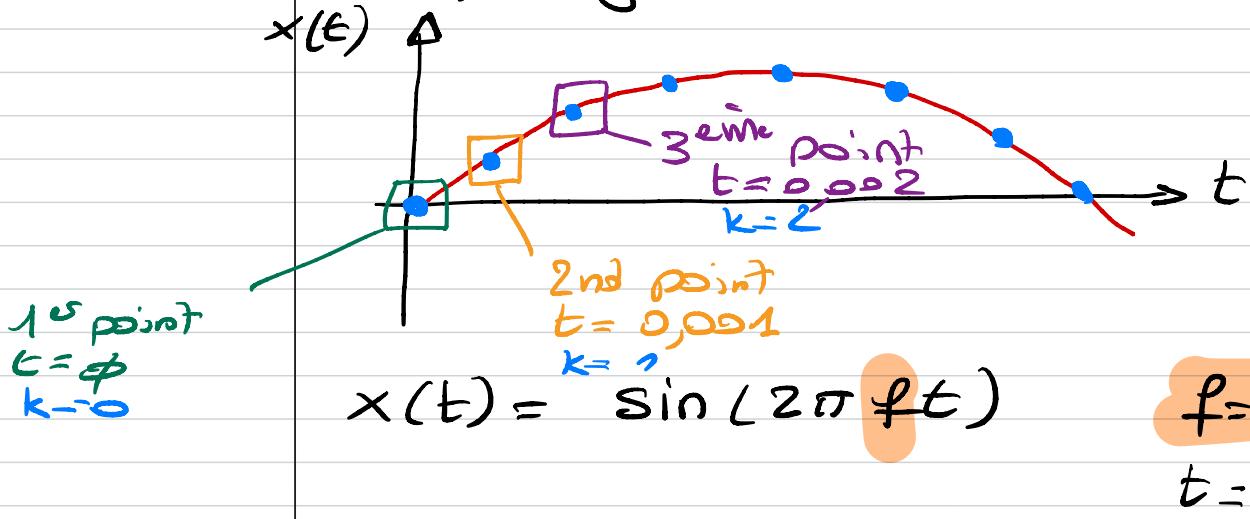
```
#define N 2000

int main(int argc, char const *argv[])
{
    double x[N];

    printf("Size of x = %lu bytes.\n", sizeof(x));
}
```

③ Remplissage du tableau x selon la formule (1) → boucle : for

Remplissage du tableau.



$$\text{step} = 0,001 \text{ [s]}$$

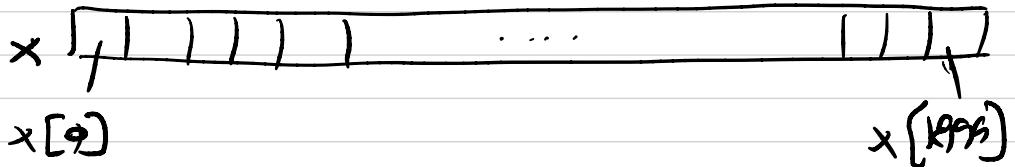
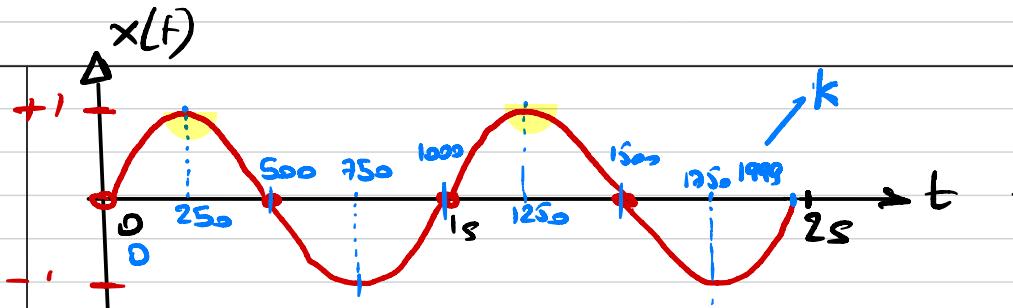
Calcul itératif des éléments du tableau.

k index pour les éléments : $[\emptyset ; N]$
ce qui correspond au temps $t : [0 ; 2]$
pas pas de 0.001 [s]

for ($k = \emptyset$; $k < N$; $k++$)

$t = k * \text{step}$;
 $x[k] = \sin \dots$

$$k_{\max} = 1999$$



④ chercher et afficher =

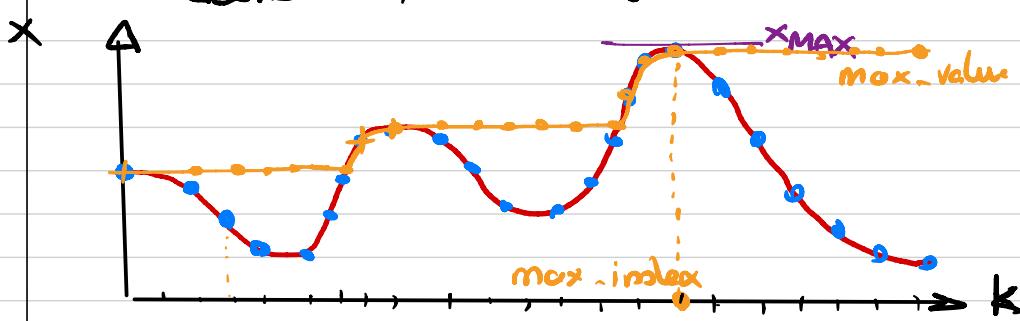
(la première)

- a - la valeur max dans x
- b - la position k correspondante.

- Algorithme ←
- Code
- Vérification

ISⁿS2.

Algorithme de recherche du max dans un tableau



l'idée: Parcourir tous les éléments du tableau x à la recherche du plus grand.

boucle for sur le tableau $k : [0; N[$

à l'index k je dois vérifier si $x[k]$ est plus grand que les valeurs précédentes

si oui → je mémorise cette valeur et la valeur k

si non examiner l'échantillon suivant

Valeur max mémorisée → max-value
index correspondant → max-index.

Pour k allant de \emptyset à $N - 1$ faire

```

    [ si  $x[k] > \text{max-val.}$  alors
        [ max-val.  $\leftarrow x[k]$ 
        max-index  $\leftarrow k$ 
        fin si;
    ]
    { Valeur de max-value
    { ou départ ?
  
```

$$\text{max-val.} = x[\emptyset];$$

$$\text{max-index} = \emptyset$$

16'03.
1C'15
=

1) Calculer la recherche de max value et max-index
2) afficher les 2 valeurs

```

// store the first value as the maximum and the related index
max_index = 0;
max_value = x[max_index]; } debut à 1 car x[0]
for (k = 1; k < N; k++) {
    if(x[k] > max_value) {
        max_value = x[k];
        max_index = k;
    }
}
printf("max value = %lf at index = %u\n", max_value, max_index);

```

⑤ Même travail pour la valeur mini et sa position dans x.

(devrait être -1.0 en k=750)