## Recherche d'un élément dans un tableau

### Recherche séquentielle :

Soit T un tableau de N entiers et x un entier donné. Ecrire une fonction qui permet de vérifier l'existence de x dans le tableau T.

**Principe :**

Cette méthode consiste à examiner les éléments du tableau un par un, jusqu'à trouver la valeur recherchée ou atteindre la fin du tableau.

**Algorithme**

```
Fonction Existe(x : entier, T :Tab, N :entier) : booléen
Début
        i←0
        test←faux
        Tantque (test=faux) et (i<N) faire
                Si T[i]=x alors
                        test←vrai
                Sinon
                        i ←i+1
                FinSi
        FinTantque
        Retourner   test
Fin
```

**Programme en Python**

```
def Existe(x, T, N) :
        test= False
        i=0
        while test == False and i<N:
                if T[i]==x:
                        test=True
                else:
                        i=i+1
        return test
```

**Algorithme**

```
Fonction Existe (x : entier, T :Tab, N :entier) :booléen
Début
        i←0
        test←faux
        Répéter
                Si T[i]=x alors
                        test←vrai
                Sinon
                        i ←i+1
                FinSi
        Jusqu' a (test=vrai) ou (i=N)
        Retourner   test
Fin
```

**TDOL**

| Objet | Type/Nature |
|---|---|
| i | Entier |
| test | Booléen |

### Recherche de la position de x dans le tableau T

```
Fonction chercher(x : entier, T : Tab, N : entier) : entier
Début
    i←0
   Tantque i<N et T[i] ≠ x faire
        i←i+1
   FinTantque
   Retourner i
Fin
```

```
def chercher(x, T, N) :
        i=0
        while i<N  and T[i]!=x:
                        i=i+1
        return i
```

**Remarque :**

**Si chercher(x, T, N) = N alors l'élément ( x ) n'existe pas dans le tableau T**

### Recherche dichotomique :

Maintenant, on suppose que le tableau T est trié dans l'ordre croissant. Ecrire une fonction qui permet de vérifier l'existence de x dans le tableau T.

Si un tableau T est trié, alors il existe un moyen très efficace pour chercher un élément x dans le tableau : il suffit d'appliquer l'algorithme de recherche dichotomique.

**Principe :**

- Trouver la position la plus centrale du tableau (si le tableau est vide, sortir)
- Comparer la valeur de cette case à l'élément recherché x. Trois cas se présentent :
  - Si x est égale à la valeur centrale, alors renvoyer la position
  - Si x est strictement inférieure à la valeur centrale, alors reprendre la procédure dans la moitié gauche du tableau.
  - Si x est strictement supérieure à la valeur centrale, alors reprendre la procédure dans la moitié droite du tableau

**Algorithme**

```
Fonction Existe (x : entier, T :Tab, N :entier) : booléen
Début
        g←0
        d← N-1
        test←faux
        Tantque (test=faux) et (g ≤ d) faire
              m←(g+d) div 2
              Si T[m] = x alors
                    test←vrai
              Sinon si T[m] > x alors
                    d← m-1
              Sinon
                    g← m+1
              Fin si
        FinTantque
        Retourner  test
Fin

Fonction Existe (x : entier, T :Tab, N :entier) : booléen
Début
        g←0
        d← N-1
        test←faux
        Répéter
              m←(g+d) div 2
              Si T[m] = x alors
                    test←vrai
              Sinon si T[m] > x alors
                    d← m-1
              Sinon
                    g← m+1
              Fin si
        Jusqu'à (test=vrai) et (g > d) faire
        Retourner  test
Fin
```

**Programme en Python**

```
def Existe(x, T, N):
    g = 0
    d = N-1
    test = False
    while (test== False) and (g <= d) :
      m = (g + d) // 2
      if  x == T[m] :
         test = True
      elif x < T[m] :
         d = m-1
      else :
         g = m+1
    return test
```

**TDO :**

| Objet | Type/Nature |
|---|---|
| g, d, m | Entier |
| test | Booléen |