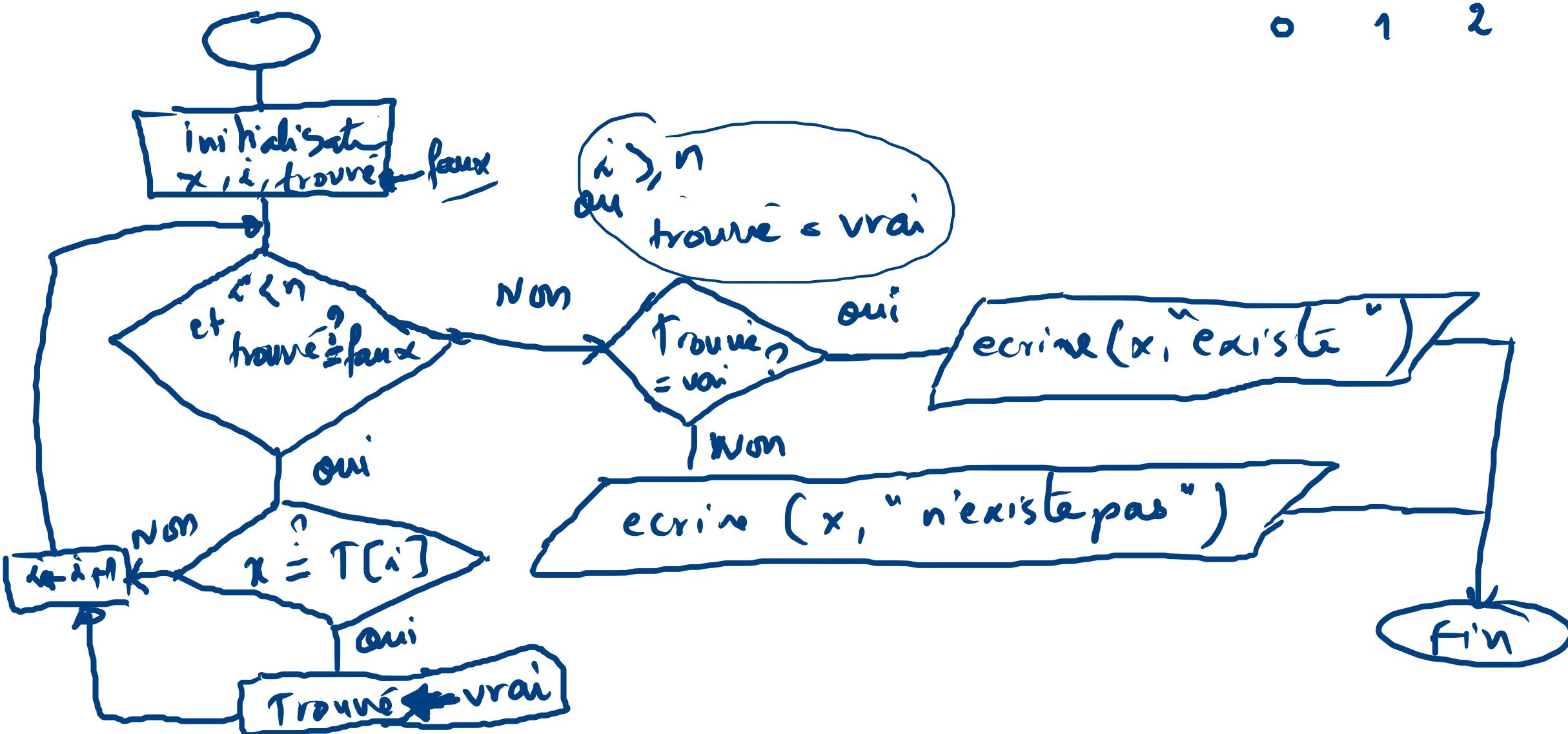
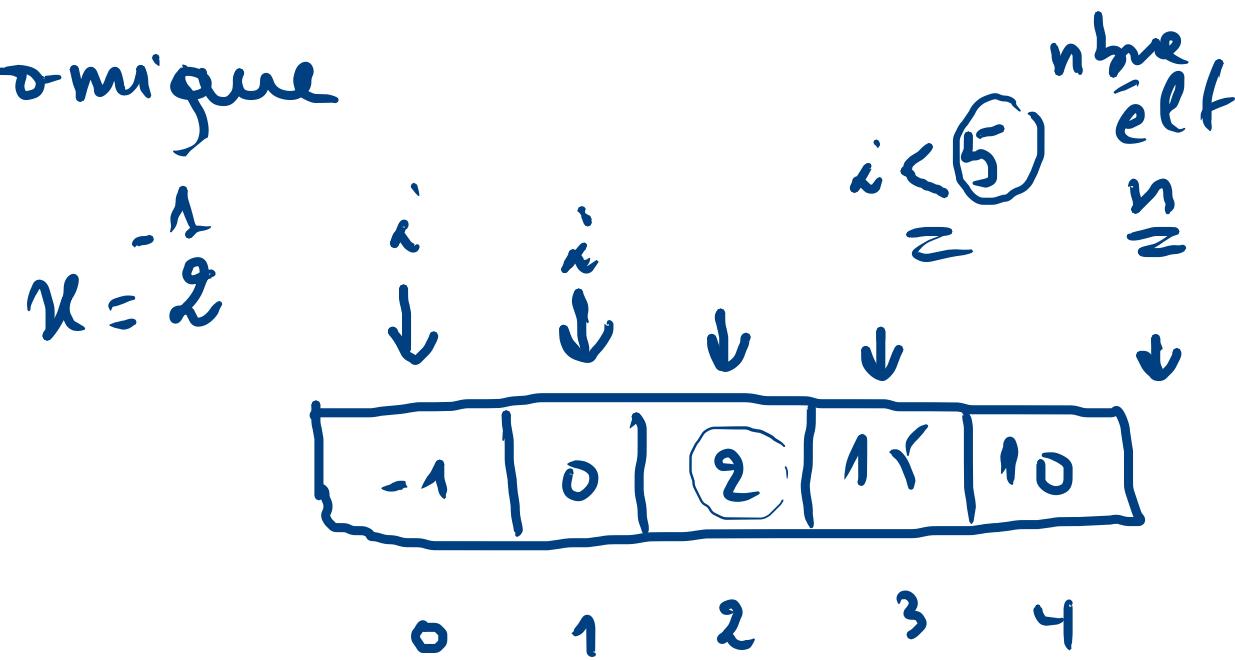


# Algorithmes de recherche

- \* La recherche séquentielle dichotomique

1) la recherche séquentielle

→ { Tableau  
elt recherché



## Algorithme : recherche\_Sequentielle

$T[0\dots N-1]$  : un tableau rempli d'entier ;

$x$  : entier ; // elt recherché

$i$  : entier ; // pour parcourir le tableau

trouvé : booléen ; // drapeau pour marqué l'elt trouvé

Début

// initialisati

$\rightarrow i \leftarrow 0;$   
 $\rightarrow \text{trouvé} \leftarrow \text{faux};$  // Considérer que l'elt n'existe pas

Tant que ( $i < n$  et  $\text{trouvé} \neq \text{faux}$ ) faire :

}     Si ( $T[i] \stackrel{?}{=} x$ ) alors

        trouvé  $\leftarrow$  vrai;

    fin Si

$i \leftarrow i + 1;$

Fin Tant que

$\rightarrow$  Si ( $\text{trouvé} = \text{vrai}$ ) alors écrive ( $x$ , "existe");

    Sinon écrive ( $x$ , "n'existe pas");

Fin

## Algorithme : recherche\_Sequentielle

T[0...N-1] : un tableau rempli d'enher;  
x : entier ; // est recherché  
i : entier; // pour parcourir le tableau  
trouvé : booléen; // drapeau pour marqué l'est trouvé

Début

// initialisation

→  $i \leftarrow 0;$  // trouv<sup>e</sup> = faux ; // Considère que l'elt n'existe pas

Tant que ( $i < n$  et trouv<sup>e</sup>  $\neq$  faux) faire :

    Si (T[i]  $\stackrel{?}{=}$  x) alors

        trouv<sup>e</sup>  $\leftarrow$  vrai;

    fin Si

    i  $\leftarrow i + 1;$

fin Tant que

    Si (trouv<sup>e</sup> = vrai) alors écrive (x, "existe");

    Sinon écrive (x, "n'existe pas");

fin

#include < stdio.h >

void main( ) {

    float T[] = { 0.5, 3.2, ..., 10.2 };

    float x;

    int i, trouve, N = 10;

    i = 0;

    trouve = 0; // O(E) faux

    printf("enher la valeur  
recherchée: ");

    scanf("%f", &x);

    while (i < N && !trouve) {

        if (T[i]  $\stackrel{?}{=}$  x)

            trouve = 1;

        i++;

    } (trouve) // if (trouve != 0)  
    printf("%f existe, x);

else printf("%f n'existe pas", x);

			$\downarrow$	$\downarrow$	
12	0	15	12	13	-1

6 élfs  
 $n = 6$        $\neg \exists (i < n)$

~~$x \neq \text{live}(x)$~~   $\Rightarrow x = 14$       trouve = faux

$i = 0$        $i < n$  et non  $\text{live}(T[i]) \leq 14 \Rightarrow 12 \leq 14$  vrai  $\Rightarrow i \leftarrow i + 1$

$i = 1$       on      "       $\neg 0 \leq 14 \Rightarrow \text{faux} \Rightarrow i \leftarrow i + 1$

$i = 2$       on      "       $\neg 15 \leq 14 \Rightarrow \text{faux} \Rightarrow \text{faux}$

$i = 3$       on      "       $\neg 12 \leq 14 \Rightarrow \text{vrai} \Rightarrow \text{faux} \Rightarrow \text{faux}$

$i = 4$       on      "       $\neg 13 \leq 14 \Rightarrow \text{vrai} \Rightarrow \text{faux} \Rightarrow \text{faux}$

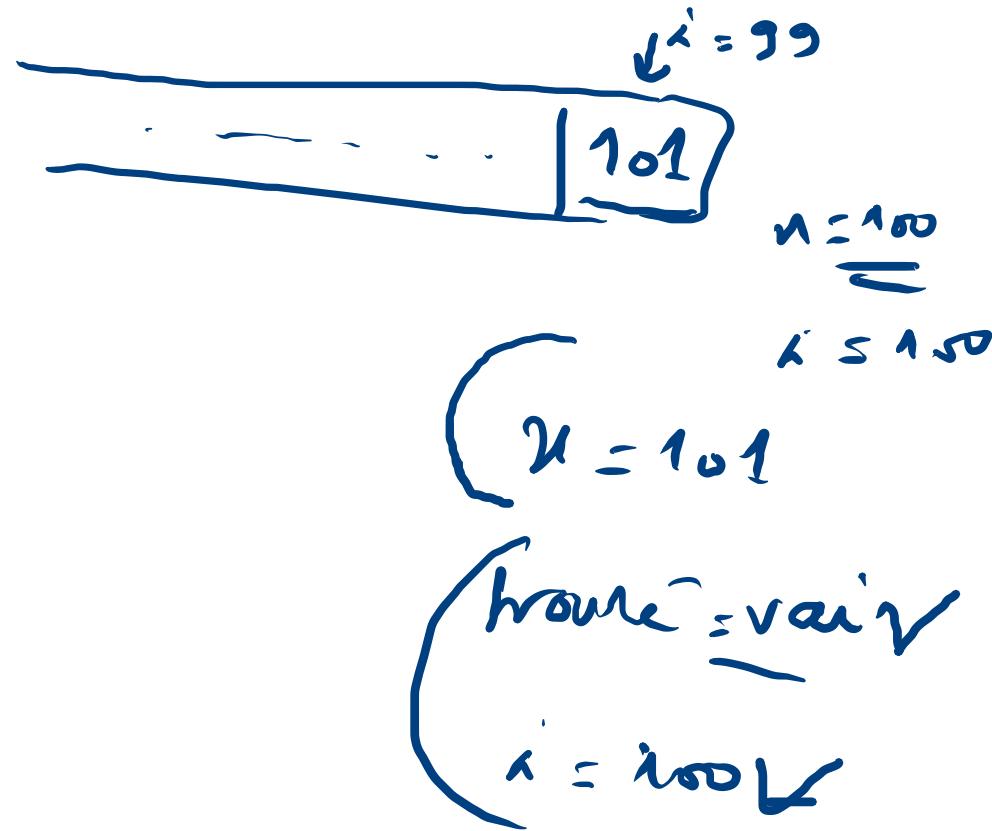
$i = 5$       on      "       $\neg -1 \leq 14 \Rightarrow \text{vrai} \Rightarrow \text{faux} \Rightarrow \text{faux}$

$i = 6$

$i \geq 6$   
 Non  
 trouve = vrai  $\Rightarrow \text{non}$   $\Rightarrow$  l'elf n'existe pas

$\Downarrow$   
 $i > n$

Rebut  
 $\text{line}(n);$   
 $i \leftarrow 0;$   
 trouvez & faux;  
 répéter :  
 si  $T[i] = \infty$ . Alors  
 { trouvez'  $\leftarrow$  trouvez j  
 fin si  
 $i \leftarrow i + 1;$   
 jusqu'à  $i > n$ . Si trouvez' = vrai ),  
 si ( trouvez' = vrai ) Alors  
 écrire ( $\infty$ , "existe");  
 sinon  
 écrire ( $\infty$ , "n'existe pas");  
 fin si  
 Fin.



## 2) recherche dichotomique

→ Tableau trié

→ l'elt

recherché  $x = 3$

$\inf = 0$		$\sup \downarrow$	$3 \downarrow$			$\sup = 6$
-1	3	100	201	999	1000	2020

$$\text{milieu} = (\inf + \sup)/2 = (0+6)/2 = 3$$

Division entière

$\inf \leq \sup$

Comparer  $T[\text{milieu}] \stackrel{?}{=} x \Rightarrow$

s'arrêter  
du programme

Sinon

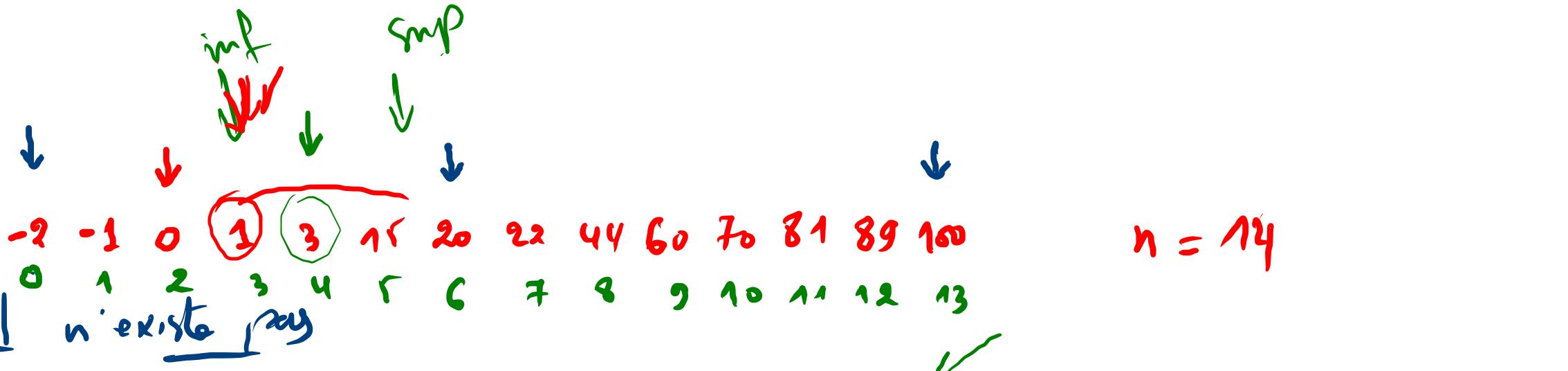
si  $T[\text{milieu}] > x$

$$\sup = \text{milieu} - 1$$

Sinon

$$\inf = \text{milieu} + 1$$

$$\text{milieu} = (0 + 2)/2 = 1$$



$\text{sup} \leq \text{inf}$

inf	sup	milieu
0	13	$6,5 \Rightarrow 6$
3	5	$2,5 \Rightarrow 2$
inf = milieu + 1		$(3+5)/2 = 4$
3		$\text{sup} = \text{milieu} - 1$
		$\text{sup} = 3$

$$T[\text{milieu}] = n$$

$\frac{n}{2}$

$$0 = 2 \frac{n}{2}$$

$$3 = 2 \frac{n}{2}$$

$$1 = 2 \frac{n}{2}$$

$$T[\text{milieu}] < n$$

$\frac{n}{2} < 2$

$$0 < 2 \frac{n}{2}$$

$$3 < 2 \frac{n}{2}$$

$$T[\text{milieu}] < n$$

$$1 < 2 \frac{n}{2}$$

$\text{inf} = \text{milieu} + 1$

$\text{inf} = 4$

sortie

l'elt n'existe pas

trouvé

$$\begin{cases} \inf = 0 \\ \sup = n-1 \end{cases}$$

$$x = -1$$

-1      3      4      10      11      101      elts  
 0      1      2      3      4      5      ( ← indices

$$\inf$$

$$0$$

$$\sup$$

$$5$$

$$\text{milieu}$$

$$2$$

$$1$$

$$1/2 = 0$$

$$\begin{array}{ll}
 T[\text{milieu}] \stackrel{?}{=} x & T[\text{milieu}](x) \\
 4 \stackrel{?}{=} -1 \text{ nm} & 4 < -1 \text{ nm} \\
 -1 \stackrel{?}{=} -1 \text{ nm} & \\
 \text{Trouve rien} &
 \end{array}$$

(  $\inf \leq \sup$  et trouvé = faux )

Sortie

## Algorithme : rechercheDichotomique

T[0..n-1] : Tableau plein et trié d'en bas

inf, sup, milieu : entier ; // indices

trouve : booléen

x : entier ; // l'est ce qu'il recherche

Début

lire(x);

inf ← 0; sup ← N-1;

trouve ← faux;

Tant que (inf <= sup et trouve = faux) :

    milieu ← (inf + sup) \ 2 ;

    si (T[milieu] = x) alors

        trouve ← vrai;

    sinon si (T[milieu] > x) alors

        sup ← milieu - 1;

    sinon

        inf ← milieu + 1;

    fin si

fin Tant que

    si (trouve = vrai) alors écrire ("x, \"existe"),

    (sinon écrire ("x, \"n'existe pas"));

fin

## Algorithme : rechercheDichotomique

T[0..N-1] : Tableau plein et trié d'entier

inf, sup, milieu : entier ; // indices

trouve : booléen ;

x : entier ; // l'elt recherché

Début

lire(x);

inf  $\leftarrow$  0; sup  $\leftarrow$  N-1;

trouve  $\leftarrow$  faux;

Tant que (inf  $\leq$  sup et trouve = faux) :

    milieu  $\leftarrow$  (inf + sup) \ 2;

    Si (T[milieu] = x) alors

        trouve  $\leftarrow$  vrai;

    Sinon Si (T[milieu] > x) alors

        sup  $\leftarrow$  milieu - 1;

    Sinon

        inf  $\leftarrow$  milieu + 1;

    Fin si

Fin Tant que

Si (trouve = vrai) alors ecrire ("x, "existe");

(sinon ecrire ("x, "n'existe pas"));

Fin si

Fin

implémentation en C :

(voir TP).