

---

## Algorithmique II

### Examen de rattrapage

Durée : 1h 30mn

#### Exercice 1 : (Sur 8 points)

La procédure *Tri\_Simple* décrite ci-dessous est probablement la méthode de tri, la plus simple à comprendre et à programmer. Elle est donnée par :

Soit  $T[1..N]$  un tableau d'entiers à trier par ordre croissant.

On parcourt le tableau  $T$  jusqu'à rencontrer deux éléments successifs qui ne sont pas dans le bon ordre. On permute alors ces deux éléments et on revient au début du tableau pour refaire le même travail. On continue ainsi, jusqu'à ce qu'on atteigne la fin du tableau  $T$ , ce dernier sera alors trié par ordre croissant.

- A) Ecrire en pseudo-code la procédure *Tri\_Simple*.
- B) Donner les étapes d'application de la procédure *Tri\_Simple* au tableau  $T = [5, 2, 3, 6, 8, 7]$ .
- C) Calculer la complexité temporelle  $t(N)$  de la procédure *Tri\_Simple* dans le cas où le tableau  $T$  est déjà trié par ordre croissant.

#### Exercice 2 : (Sur 6 points)

On considère un tableau  $A$  de  $n$  entiers naturels, trié par ordre croissant, et  $x$  un entier naturel.

1. En utilisant un tableau intermédiaire  $A1$  de  $n$  entiers, écrire la procédure :  
*Chercher*( $A : \text{Entier}[1..n], x : \text{entier}$ ), dont la complexité temporelle en pires des cas est  $\theta(n)$ , qui affiche deux éléments de  $A$  dont leur somme est égale à  $x$ ; s'ils existent ; sinon elle affiche un message indiquant qu'il n'y a pas de couple d'éléments de  $A$  dont la somme est égale à  $x$ .
2. Montrer qu'effectivement la complexité temporelle dans les pires des cas de la procédure *Chercher*( $A, x$ ) est égale à  $\theta(n)$ .

**Exercice 3 : (Sur 6 points)**

On considère la procédure *Calcul* ( $T : \text{Entier}[0..N]$ ,  $deb : \text{Entier}$ ,  $fin : \text{Entier}$ ) donnée par :

**Procédure** Calcul ( $T : \text{Entier}[0..N]$ ,  $deb : \text{Entier}$ ,  $fin : \text{Entier}$ )

**Var**     $i, j : \text{Entier}$

**Debut**

$j \leftarrow deb$

**Pour** ( $i$  allant de  $deb$  à  $fin-1$ ) **Faire**

**Si** ( $T[i] \leq T[fin]$ ) **alors**

            Echanger ( $T[i]$ ,  $T[j]$ )

$j \leftarrow j + 1$

**Fin Si**

**Fin Pour**

    Echanger ( $T[fin]$ ,  $T[j]$ )

**Fin**

Où Echanger ( $x$ ,  $y$ ) est la procédure qui consiste à permuter les deux éléments  $x$  et  $y$ .

1. Faites tourner la procédure Calcul( $T$ ,  $deb$ ,  $fin$ ) sur le tableau  $T=[2, 9, 3, 8, 6, 10, 4, 5]$ , avec  $deb=0$  et  $fin=7$ .
2. Quel est le but de la procédure Calcul( $T$ ,  $deb$ ,  $fin$ ) ?

*Bon courage*