

## Contrôle continu 1

*Durée : 1 heure**Aucun document. Aucune machine. Barème indicatif. Tous les exercices sont indépendants.***Exercice 1 : exécution d'algorithme (4 points)**

Exécutez à la main le tri par insertion vu en cours sur le tableau suivant :

25	7	25	2	80
----	---	----	---	----

Combien de comparaisons avez-vous dû faire ? Ecrivez les tableaux intermédiaires à la fin de chaque itération  $i$  de la boucle extérieure ( $i \in 0, 1, 2, 3$ ). Pour répondre, remplissez la table suivante. Est-ce que le tri par insertion est stable ? Justifiez votre réponse sur le tableau donné.

**Réponse : Le tri est stable. 6 comparaisons au total. Les tableaux intermédiaires sont :**

Itérations	Tab. intermédiaires et comparaisons	Nombre
i=0	[7,25,25,2,80] (7<25)	1
i=1	[7,25,25,2,80] (25=25)	1
i=2	[2,7,25,25,80] (2<25)(2<25)(2<7)	3
i=3	[2,7,25,25,80] (80>25)	1
Total		6

**Exercice 2 : lecture d'algorithme (6 points)**

T est un tableau d'entiers de longueur n.

```

1  comp(a, b){
2    if (a>=0 && b<0)
3      renvoie true
4    else if (a<0 && b<0 && a<b)
5      renvoie true
6    else if (a>=0 && b>=0 && a<b)
7      renvoie true
8    else renvoie false
9  }
10
11  QueFaisJe(T) :
12    for i from n-1 downto 1
13      for j from 0 to i-1
14        if (comp(T[i],T[j]))
15          exchange T[i] and T[j]
```

- Exécuter l'algorithme QueFaisJe sur le tableau  $T=\{10,-2,8,7,-4,-3,-5\}$ . Que fait cet algorithme ? **Réponse : L'algorithme ordonne les éléments du tableau en mettant d'abord les entiers positifs dans l'ordre croissant, puis les négatifs dans l'ordre croissant.** Combien de comparaisons sont effectuées ? Ecrivez les tableaux intermédiaires à la fin de chaque itération  $i$  de la boucle extérieure en remplissant la table suivante.

Itérations	Tableau T	Nombre
i=6	[10,-5,8,7,-4,-3,-2]	6
i=5	[10,-5,8,7,-4,-3,-2]	5
i=4	[10,-4, 8,7,-5,-3,-2]	4
i=3	[7,10,8,-4,-5,-3,-2]	3
i=2	[7,8,10,-4,-5,-3,-2]	2
i=1	[7,8,10,-4,-5,-3,-2]	1
Total		21

2. Modifiez l'algorithme **QueFaisJe(T)** pour qu'il trie un tableau d'entiers de longueur n dans l'ordre décroissant. Écrivez la réponse dans l'espace suivant. **Réponse : Il suffit de modifier la fonction comp**

```

1 comp(a, b){
2   if (a<b)
3       renvoie true
4   else renvoie false

```

**Exercice 3 : conception d'algorithme (10 points)**

Étant donné un tableau  $T$  de  $n$  éléments, nous voulons écrire un algorithme qui permette de vérifier que les  $n$  éléments sont tous distincts.

1. Le premier algorithme doit utiliser un tri sélection : écrivez le pseudo-code du tri sélection et de l'algorithme et une description courte ci-dessous.

**Je ne redonne pas le pseudo-code du tri sélection qui est donné dans le cours.**

```
1  dist(T: tableau trie avec triSelect){
2      for i from 0 to n-2
3          if(T[i]==T[i+1])
4              return false
5      return true
6  }
```

**Description courte :**

Une fois trié les éléments égaux seront les uns à coté des autres. Il suffit de faire une boucle et vérifier qu'il n'y ait pas deux éléments égaux l'un à la suite de l'autre.

2. Nous supposons maintenant que toutes les valeurs du tableau  $T$  sont comprises entre 0 et  $n - 1$ . À l'aide d'un tableau auxiliaire  $Aux$  de taille  $n$ , changer l'algorithme pour qu'il renvoie un tableau  $S$  trié dans l'ordre croissant et dans lequel il n'y a pas de doublons. Par exemple, `triSansDoublons([3,1,4,5,3,6,7,4])` devra renvoyer `[1,3,4,5,6,7]`. Attention, le tri ne sera pas en place. Servez-vous du tableau  $Aux$  pour identifier la taille du tableau  $S$  à renvoyer et pour ne pas insérer des doublons dans le tableau  $S$ .

**Pseudo-code :**

```
1  rempAux(T, Aux){
2      tailleS=0
3      for j from 0 to n-1
4          if(Aux[T[j]]==0)
5              tailleS++
6              Aux[T[j]]=Aux[T[j]]+1
7      return tailleS
8  }
9  tabSansDoub(T, Aux)
10     tailleS=rempAux(T, Aux)
11     S= new Tableau int[tailleS]
12     l=0
13     for i from 0 to m
14         if (Aux[i]!=0){
15             S[l]=i
16             l++
17         }
18     return S
```

**Description courte :**

On peut supposer plus généralement les éléments de  $T$  compris entre 0 et  $m$ , et on peut alors les associer avec les indices d'un tableau auxiliaire qui va de 0 à  $m$ . Le tableau  $Aux$  contiendra, à la place  $i$ , le nombre d'occurrences de l'entier  $i$  dans  $T$ . On pourra au même temps tenir un compteur qui nous dira quel est la taille du nouveau tableau  $S$  à renvoyer. Il suffira de parcourir le tableau  $Aux$  et remplir  $S$  avec les éléments de  $Aux$  différent de 0. Les éléments de  $S$  seront triés.