

**Examen d'algorithmique
aucun document, durée 2h**

1. Algorithme à tester

Q1) (5 pts) Vérifiez en prenant l'exemple donné si le déroulement de l'algorithme de la procédure Tri() permet d'obtenir le résultat attendu. Pour cela vous présenterez vos variables sous la forme d'un tableau (voir dernière page) et vous expliquerez votre raisonnement. (Il n'est peut être pas nécessaire de détailler toutes les lignes à chaque fois !!!).

1 :	PROCEDURE Tri(ES tableau T () : réel, E NbreCase : entier)
	VARIABLES
2 :	i : entier
3 :	compteur, memoire : réel
4 :	FinTrie : booléen
5 :	pour i ← 2 à NbreCase pas 1
6 :	memoire ← T(i)
7 :	compteur ← (i – 1)
8 :	Répéter
9 :	FinTrie ← vrai
10:	Si T(compteur) > memoire alors
11:	T(compteur + 1) ← T(compteur)
12:	compteur ← compteur-1
13:	FinTrie ← faux
14:	FSi
15:	Si compteur < 1 alors
16:	FinTrie ← vrai
17:	FSi
18:	Jusqu'à FinTrie = vrai
19:	T(compteur + 1) ← memoire
20:	FPour
21:	FinProcédure

Q2) (3pts) Déterminez la complexité de cet algorithme.

2. Multiplication égyptienne

Méthode :

Les égyptiens de l'antiquité savaient :

- additionner deux entiers strictement positifs,
- soustraire 1 à un entier strictement positif,
- multiplier par 1 et 2 tout entier strictement positif,
- diviser par 2 un entier strictement positif pair.

Voici un exemple qui multiplie 14 par 13 en utilisant uniquement ces opérations :

$$\begin{aligned}
 14 \times 13 &= 14 + 14 \times (13 - 1) = 14 + 14 \times 12 \\
 &= 14 + (14 \times 2) \times (12 / 2) = 14 + 28 \times 6 \\
 &= 14 + (28 \times 2) \times (6 / 2) = 14 + 56 \times 3 \\
 &= 14 + 56 + 56 \times (3 - 1) = 70 + 56 \times 2 \\
 &= 70 + (56 \times 2) \times (2 / 2) = 70 + 112 \times 1 \\
 &= 70 + 112 = 182
 \end{aligned}$$

Algorithme à rechercher : Ecrire l'algorithme récursif qui permet la multiplication de 2 naturels suivant cette méthode :

Q3) (2 pts) Déterminer le ou les cas terminaux (cas où la récursivité s'arrête).

Q4) (2 pts) Déterminer le ou les cas généraux (appel à la récursivité).

Q5) (4 pts) Donner le corps de la fonction suivante en utilisant un algorithme récursif :

Fonction MultiplicationEgyptienne(E a : entier, E b : entier) : entier

Q6) (2 pts) Question bonus : Déterminer la complexité de votre algorithme.

3. Recherche de chaînes de caractères

Q1) (2 pts) Expliquer les différences de performances entre les algorithmes

- 1) Algorithme naïf
- 2) Algorithme de Rabin-Karp

Q2) (2 pts) On vous rappelle en annexe la procédure KMP. Il faut adapter cette procédure pour récupérer dans une structure adéquate le nombre d'occurrences trouvées et pour chaque occurrence son emplacement dans le tableau T. On donnera une solution sous forme de procédure et une autre sous forme de fonction.

Rappel sur la déclaration de structures

```

type nomStructure = structure
    nomChamp1 : typeChamp1
    ...
    nomChampn : typeChampn
fin structure
  
```

