

EXAMEN

Section: LFSI1

Épreuve d': Algorithmique et structures de données I

Nature de l'épreuve:	DC <input type="checkbox"/> DS <input type="checkbox"/> EF <input checked="" type="checkbox"/>	Documents:	autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/>
Date de l'épreuve:	20/06/2015	Calculatrice:	autorisée <input checked="" type="checkbox"/> non autorisée <input type="checkbox"/>
Durée de l'épreuve:	1H30	Session:	principale <input type="checkbox"/> contrôle <input checked="" type="checkbox"/>

Exercice 1 (6 pts)

N.B : Résoudre cet exercice sans utiliser les tableaux.

1. Ecrire une fonction **SAISIE** permettant de saisir un entier strictement positif.
2. Ecrire une fonction **CONVERSION** permettant de retourner le résultat de la conversion d'un nombre décimal en un nombre binaire. La conversion se fait par des divisions successives par 2 jusqu'à obtenir un quotient nul, le résultat est la séquence des restes dans l'ordre inverse.
3. Ecrire un **algorithme principal** permettant de saisir une suite d'entiers strictement positifs qui se termine par 999 et afficher leurs équivalents en binaire.

Exercice 2 (7 pts)

N.B : Résoudre cet exercice en utilisant le formalisme pointeurs et l'allocation dynamique

1. Ecrire une fonction **SAISIE** permettant de lire la dimension d'un tableau.
2. Ecrire une procédure **LECTURE** permettant de remplir un tableau par des entiers.
3. Ecrire une procédure **AFFICHAGE** permettant d'afficher les éléments d'un tableau.
4. Ecrire une procédure **REORGANISATION** permettant de réorganiser un tableau de manière que les apparitions d'une valeur soient consécutives sans utiliser un autre tableau.
5. Ecrire un **algorithme principal** faisant appel aux fonctions et procédures ci-dessus pour remplir un tableau T par N entiers, le réorganiser et l'afficher.

Exemple :

Donner la dimension de T : 10

T :

7	6	5	9	9	5	7	4	7	9
---	---	---	---	---	---	---	---	---	---

Résultat :

T :

7	7	7	6	5	5	9	9	9	4
---	---	---	---	---	---	---	---	---	---

Exercice 3 (7 pts)

1. Ecrire une procédure **SAISIE_CH** permettant de saisir une chaîne CH formée de lettres minuscules de longueur inférieure ou égale à 20.
2. Ecrire une procédure **SAISIE_TAB_CH** permettant de :

- Lire la taille d'un tableau TAB_CH (N est pair et compris entre 2 et 30)
 - Remplir TAB_CH par N mots dont chacun est non vide, formé par des lettres minuscules et de longueur inférieure ou égale à 20.
3. Ecrire une fonction **PALINDROME** qui retourne 1 si une chaîne CH (passé en paramètre) est palindrome, 0 si non. Un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche.
 4. Ecrire une procédure **ALTERNANCE** permettant de concaténer, en alternance, deux chaînes CH1 et CH2 dans une troisième chaîne CH3, c'est à dire on prend en premier le premier caractère de CH1, ensuite le premier caractère de CH2, par la suite le deuxième caractère de CH1, le deuxième caractère de CH2 et ainsi de suite jusqu'à parcours complet des deux chaînes. Si une chaîne se termine avant l'autre, on ajoute à CH3 tous les caractères restants de l'autre chaîne.
 5. Ecrire une procédure **TRI_TAB_CH** permettant de trier les chaînes contenues dans un tableau de chaînes TAB_CH dans l'ordre décroissant selon leurs tailles.
 6. Ecrire une procédure **AFFICHER** permettant d'afficher un tableau de chaînes TAB_CH.
 7. Ecrire un **algorithme principal** permettant de :
 - Remplir un tableau TAB_CH par N chaînes
 - Afficher les mots palindromes contenus dans TAB_CH.
 - Trier le tableau TAB_CH.
 - Concaténer, en alternance, les mots de TAB_CH deux à deux de la manière suivante : le plus long mot avec le plus court mot, le deuxième plus long mot avec le deuxième plus court mot et ainsi de suite. Afficher les résultats des concaténations.
 - Afficher le tableau TAB_CH.