Licence Sciences et Technologies	**ANNEE : 2005/2006** **SESSION DE SEPTEMBRE 2006** **ETAPE :** INF2 MAT2 MAI4 **UE :** INF102 **Epreuve :** Initiation à l'algorithmique **Date :** 6 Septembre 2006 **Heure :** 11 H 00 **Durée :** 1 H 30 **Documents :** Tous documents interdits **Vous devez répondre directement sur le sujet qui comporte 4 pages.** **Insérez ensuite votre réponse dans une copie d'examen comportant tous les renseignements administratifs** **Epreuve de** Mme Delest.

Indiquez votre code **d'anonymat** : N° : .

La notation tiendra compte de la clarté de l'écriture des réponses.

Barème

- Question 1 – Connaissances générales : 4 points
- Question 2 – Compréhension d'algorithme : 3 points
- Question 3 – Compréhension d'algorithme récursif : 2 points
- Question 4 – Ecriture de fonctions : 3 points
- Question 5 – Algorithme de tri : 4 Points
- Question 6 – Manipulation de listes : 4 points

Question 1. Cochez les affirmations qui sont correctes :

☐ La boucle *tant que* ne comporte pas de test
☐ Les instructions d'une boucle *pour* sont toujours exécutées
☐ Pour un tableau, l'accès aux éléments s'effectue directement par la clé.
☐ Les méthodes diviser pour régner consistent à appliquer le même algorithme récursivement.
☐ La complexité maximale d'un algorithme de tri est en $O(n^2)$
☐ Dans le tri fusion, les k premiers éléments du tableau sont à leur place après k appels récursifs
☐ Une liste est une table d'association de taille non constante.
☐ Dans une liste doublement chaînée de n éléments, l'accès à l'avant-dernier élément se fait en $O(n)$

Question 2. Soit la fonction suivante :

```
fonction mystere(ref T:tableau[1..N] d'entier,val N :entier) : entier ;
    var k,itmp,max,lmax :entier ;
    début
        max=1 ;
        itmp=1 ;
        lmax=1 ;
        pour k allant de 2 à N faire
            si T[k]== T[k-1] alors
                lmax=lmax+1
            sinon
                si lmax>max alors
                    itmp=k-1 ;
                    max=lmax
                finsi
                lmax=1 ;
            finsi
        finpour
        si lmax>max alors
            T[1]=T[N] ;retourner(lmax)
        sinon
            T[1]=T[itmp] ;retourner(max)
        finsi
    fin ;
finfonction
```

1. On considère le tableau A de dimension 6 contenant la séquence 3,2,5,4,6,1. Quel est le résultat de l'appel mystere(A,6) ? Quel est l'effet de cet appel sur le tableau A ?

2. Quelle sera la valeur entière maximale retournée par la fonction si le tableau lors de l'appel a pour taille N? Illustrez par un exemple. Le tableau peut-il être modifié dans ce cas? Donnez un exemple pour lequel le tableau serait modifié.

3. Que se passe-t-il si on remplace ref par val dans l'en-tête de la fonction ?

4. Donnez la complexité de cette fonction en nombre de tests ? ……

Question 3. Soit la fonction suivante définie pour N>2:

```
fonction recurse(ref T:tableau[1..N] d'entier ; val N,p entier) : entier ;
    fonction recurseRec(ref T:tableau[1..N] d'entier,val k :entier) : entier ;
        début
            si (k==N+1) alors
                retourner(0) ;
            sinon
                si T[k]==p*T[k-1] +T[k-2] alors
                    retourner(1+recurseRec(T,k+1)) ;
                sinon
                    retourner(recurseRec(T,k+1))
                finsi
            finsi
        fin
    finfonction ;
    début
        retourner (recurseRec(T,3)) ;
    fin ;
finfonction
```

1. On considère le tableau A de dimension 5 contenant la séquence 1,1,2,3,5. Quel est le résultat de l'appel recurse(A,5,1) ? ……
2. Donner la suite des appels récursifs provoqués par l'appel de la question précédente ainsi que l'état du contexte de la fonction à chaque appel.

Question 4. Soit un tableau B d'entiers de taille N.

1. Ecrire la fonction *poc* qui prend pour paramètre un index *K* et le tableau *B* et dont le résultat est *vrai* si *B[K] = B[K-1]+B[k-2]* ou si *B[K] = B[K-1]* .

2. Ecrire une fonction *comptePoc* qui utilise la fonction *poc* et dont le résultat est le nombre de fois où un entier est égal à la somme des deux entiers qui le précèdent ou à l'entier qui le précède dans le tableau. Par exemple si le tableau contient (0,1,0,0,0,0,1,1,0,1), le résultat est 5.

Question 5. Soit un tableau T de dimension N de structures *Info* définies de la manière suivante :

Info :structure
Taille :entier ;
Poids :entier ;
finstructure

Par exemple, pour N=3, un exemple de tableau T est :

	Taille	Poids
T[1]=	150	60
T[2]=	185	95
T[3]=	150	45

Donnez les mofifications à apporter à l'une des fonctions de tri décrite dans le cours, pour qu'elle reçoive en entrée un tel tableau et dont le résultat sera ce même tableau trié suivant le champ *Taille* et en cas d'égalité trié suivant le champs *Poids*.

Question 6.

1. Soit L une liste doublement chaînée d'entiers.

   - Que contient L et P après la séquence d'instructions suivante

```
var L :listeSC d'entier ;
var P :^entier ;
creerListe(L) ;
insererEnTete(6,L) ;
insererEnTete(8,L) ;
P=premier(L) ;
insererAprès (7,L,P) ;
P=premier(L) ;
insererAprès(4,L,P) ;
P=suivant(L,P) ;
supprimerAprès(L,P) ; ;
P=premier(L) ;
supprimerAprès(L,P) ;
P=suivant(L,premier(L)) ;
```

P=……
Contenu(P)=…….
L=……

   - Ecrire une fonction *comptepocListe* dont le résultat est le nombre de fois où un entier est égal à la somme des deux entiers qui le précèdent ou à l'entier qui le précède dans la liste L. On s'inspirera de la question 4.

FIN