Licence **Sciences et Technologies**	**ANNEE : 2004/2005** **SESSION DE SEPTEMBRE 2005** **ETAPE :** INF2, MAT2 et MAI4 **UE :** INF102 **Epreuve :** Initiation à l'algorithmique **Date :** **Heure :** 11 H **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 de tris : 3 points
- Question 3 – Compréhension d'algorithme récursif : 3 points
- Question 4 – Manipulation de tableaux : 6 points
- Question 5 – 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
¤ Un tableau passé par *référence* à une fonction n'est pas recopié dans l'environnement de la fonction.
¤ Une fonction *récursive* comporte au moins une boucle
¤ L'algorithme de tri qui s'exécute toujours en $O(n)$ est le tri à bulle
¤ Dans le tri séléction, les k premiers éléments du tableau sont à leur place définitive après k itérations.
¤ Les méthodes « diviser pour régner » permettent toujours de diminuer la complexité.
¤ Dans une liste simplement chaînée, l'ajout d'un élément peut se faire en tête

Question 2. Soit la fonction suivante :

```
fonction mystere(ref T:tableau[1..NMax] d'entier,val N :entier) : entier ;
var i, k,ntmp,itmp :entier ;
var tmp :tableau[1..N*N]d'entiers ;
début
   init(tmp,0) ;
   ntmp=0 ;
   pour i allant de 1 à N-1 faire
      pour k allant de i+1 à N faire
         si T[i] >T[k] alors
            ntmp=ntmp+1 ;
            tmp[ntmp]=T[k] ;
         finsi ;
      finpour ;
   finpour;
   copie(tmp,T,1,ntmp ,1);
   retourner(ntmp) ;
fin ;
finfonction
```

où *init* et *copie* sont les fonctions décrites dans le cours.

1. On considère le tableau A de dimension 8 contenant la séquence 3,6,5,1,2,4,7. Quel est le résultat de l'appel mystere(A,7) ? Quel est l'effet de cet appel sur le tableau A ?

2. Quelle sera la valeur entière maximale (en fonction de N) retournée par la fonction mystere ? Illustrez par un exemple.

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 :

```
fonction recurse(ref T:tableau[1..N] d'entier) :  entier ;
   fonction recurseRec(ref T:tableau[1..N] d'entier,val i :entier) : entier ;
      début
         si i<N  alors
            si T[i]>T[N] alors
               retourner(1+recurseRec(T,i+1)) ;
            sinon
               retourner(recurseRec(T,i+1)) ;
         finsi
      fin
   finfonction ;
   début
      recurseRec(T,1) ;
   fin ;
finfonction
```

1. On considère le tableau A de dimension 4 contenant la séquence 3,4,1,2. Quel est le résultat de l'appel recurse(A) ? ……

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.

3. Le tableau est passé par référence. S'il était passé par valeur, quel serait l'effet sur la complexité en nombre d'opérations ? la complexité en mémoire?

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

1. Ecrire la fonction *compare* qui prend pour paramètre deux index *I* et *J,* le tableau *B* et dont le résultat est *vrai* si *B[I]>B[J]*.

2. Ecrire une fonction *compteSup* qui utilise la fonction *compare* et dont le résultat est le nombre de fois où dans un tableau B, pour deux indices *I*,*J* tels que *I<J,* on a *B[I]>B[J]*. Par exemple si le tableau contient (3,4,1,2), le résultat est 4 ((3>1), (3>2), (4>1), (4>2)).

3. Ecrire une fonction *inversion* dont le résultat est un tableau de taille maximale N*N, qui contient l'ensemble des valeurs *B[I]* pour lesquelles il existe un indice *J* tel que *I<J* et *B[I]>B[J]*. On utilisera la fonction *compare.* Par exemple la séquence (4,1,6,7,3,8,2,5) conduit au tableau (4,4,4,6,6,6,7,7,7,3,8,8).

Question 5.

1. Quelles différences y-a-t-il entre la suppression d'un élément dans une simplement liste chaînée et la suppression dans une liste doublement chaînée ? On illustrera par un exemple.

2. Soit L une liste doublement chaînée d'entiers. Que contiennent L et P après la séquence d'instructions suivante

```
var L :listeDC d'entier ;
var P :^entier ;
creerListe(L) ;
ajouterEnTete(7,L) ;
ajouterEnTete(16,L) ;
P=suivant(L,premier(L)) ;
ajouterAprès(4,L,P) ;
P=dernier(P);
ajouterAprès(12,L,P) ;
P=precedent(dernier(L)) ;
supprimer(L,P) ;
P=suivant(L,premier(L)) ;
```

P=.....
L=.....

3. Ecrire une fonction *compteSupliste* dont le résultat est le nombre de fois où dans une liste L simplement chaînée, le premier élément est supérieur à un de ses suivants. Par exemple si L=(3,4,1,2) alors le résultat est 2 (car 3>1 et 3>2) .

FIN