Licence Sciences et Technologies	**ANNEE : 2005/2006** **SESSION DE MAI 2006** **ETAPE :** INF2 MAT2 MAI4 **UE :** INF102 **Epreuve :** Initiation à l'algorithmique **Date :** 22 Mai 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 de tris : 3 points
- Question 3 – Compréhension d'algorithme récursif : 3 points
- Question 4 – Ecriture de fonctions : 3 points
- Question 5 – Manipulation de tableau : 3 Points
- Question 6 – Manipulation de listes : 4 points

Question 1. Cochez les affirmations qui sont correctes :

▣ La boucle *pour* comporte un test
▣Les instructions d'une boucle *répéter* sont toujours exécutées
☐ Une variable passée par *référence* à une fonction est recopiée dans l'environnement de la fonction.
☐ Les méthodes diviser pour régner permettent toujours d'obtenir une meilleure complexité en temps
▣ La complexité minimale d'un algorithme de tri est en *O(n)*
☐ Dans le tri rapide, les *k* premiers éléments du tableau sont à leur place après *k* appels récursifs
☐ Les listes ont toujours une complexité mémoire plus élevée que les tableaux.
▣ Dans une liste simplement chaînée de *n* éléments, l'accès au dernier élément se fait en *O(n)*

Question 2. Soit la fonction suivante :

```
fonction mystere(ref T:tableau[1..NMAX] d'entier,val N :entier) : entier ;
    var k,itmp,compt :entier ;
    var tmp :tableau[1..N]d'entiers ;
    début
        init(tmp,0) ;
        k=2;
        itmp=0;
        compt=0 ;
        tant que k<=N faire
            si T[k] ==T[k-1]+1 alors
                itmp=itmp+1 ;
                tmp[itmp]=k ;
                compt=compt+1;
            finsi ;
            k=k+1 ;
        fintantque;
        copie(tmp,T,1,itmp ,1);
        retourner(compt) ;
    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 0,0,1,0,1,1,0,1. Quel est le résultat de l'appel mystere(A,8) ? Quel est l'effet de cet appel sur le tableau A ?
   Le résultat de l'appel mystere(A,8) est 3. Le tableau A sera modifié. Il contiendra 3,5,8,0,1,1,0,1.

2. Quelle sera la valeur entière maximale retournée par la fonction si le tableau lors de l'appel a pour taille 2N et ne contient que des valeurs de l'ensemble {0,1}? Illustrez par un exemple.
   La valeur maximale sera N. En effet, le maximum sera obtenu quand on a une séquence de 0,1 qui se suivent. Par exemple, si N=6 on a le maximum pour 0,1,0,1,0,1.

3. Que se passe-t-il si on remplace ref par val dans l'en-tête de la fonction ?
   Le tableau est recopié dans la fonction. Par suite la fonction effectue une copie locale à la fonction et le tableau n'est pas modifié.

4. Donnez la complexité de cette fonction en nombre de tests ? …O(N) …….

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 k :entier) : entier ;
        début
            si (k==N+1) alors
                    retourner(0) ;
            sinon
               si T[k]==T[k-1]+1 alors
                  retourner(1+recurseRec(T,k+1) ;
            sinon
                  retourner(recurseRec(T,k+1)) ;
            finsi
        fin
    finfonction ;
    début
        retourner (recurseRec(T,2)) ;
    fin ;
finfonction
```

1. On considère le tableau A de dimension 4 contenant la séquence 0,0,1,1. Quel est le résultat de l'appel recurse(A) ? …*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.

	T	*k*	*Val. Fonct.*
recurseRec(T,2)	*@A*	*2*	
recurseRec(T,3)	*@A,@A*	*2,3*	
recurseRec(T,4)	*@A,@A,@A*	*2,3,4*	
recurseRec(T,5)	*@A,@A,@A,@A*	*2,3,4,5*	
→0	*@A,@A,@A*	*2,3,4*	*0*
→0	*@A,@A*	*2,3*	*0*
→1	*@A*	*2*	*1*
→1			*1*

3. Dans la fonction récursive, le tableau est passé par référence. S'il était passé par valeur, quel serait l'effet sur la complexité ?
   Si le tableau est passé par valeur, la complexité augmente en temps (recopie du tableau à chaque appel récursif) et en mémoire (place prise par la copie du tableau à chaque appel récursif).

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

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

```
fonction pic(ref B :tableau[1..N]d'entiers ;val k :entier) :booléen ;
  début
    retourner((k>1) et (k<=N) et (B[K] = = B[K-1]+1))
  fin
finfonction
```

2. Ecrire une fonction *comptePic* qui utilise la fonction *pic* et dont le résultat est le nombre de fois où un entier est égal à l'entier qui le précède plus 1 dans le tableau. Par exemple si le tableau contient (0,1,0,0,1,0,1,1,0,1), le résultat est 4.

```
fonction comptePic(ref B :tableau[1..N]d'entiers) :entier ;
  var i,c :entiers ;
  début
    c=0 ;
   pour i allant de 2 à N faire
     si pic(B,i) alors c=c+1 finsi;
   finpour
   retourner(c)
  fin
finfonction
```

Question 5. Soit un tableau T de dimension M de tableaux de taille N d'entiers triés en ordre croissant . Par exemple, M=3, N=4, un exemple de tableau T est :

T[1]=	2	5	6	12
T[2]=	1	3	7	10
T[3]=	5	6	9	14

En utilisant la fonction *fusion* décrite dans le cours, écrire une fonction *multiTri* qui reçoit en entrée un tel tableau et dont le résultat est un tableau d'entiers à une seule dimension contenant tous les entiers du tableau T triés en ordre croissant. On ne demande pas de réécrire la fonction *fusion*. On rappelle l'en-tête de cette fonction décrite en cours :

```
fonction fusion(ref T1:tableau[D1..N1] d'entier; ref T2:tableau[D2..N2] d'entier;
                val DT1,FT1,DT2,FT2:entier):tableau[1..FT1+FT2-DT1-DT2+2] d'entier;
fonction multiTri(ref T :tableau[1..M] de tableau[1..N] d'entiers) :tableau[1..N*M]d'entiers ;
  var R : tableau[1..N*M]d'entiers ;
  var i,d : entier ;
  fonction fusionBis(ref A :tableau[1..N]d'entier ;ref B :tableau[1..N*M]d'entier ;
                     val p :entier ;val q :entier) :vide ;
    var F :tableau[1..p+q] d'entiers ;
    début
      F=fusion(A,B,1,p,1,q) ;
      copie(F,R,1,p+q,1) ;
    fin
  finfonction
 début
   copie(T[1],R,1,N,1) ;
   d=N ;
   pour i allant de 2 à M faire
     fusionBis(T[i],R,N,d) ;
     d=d+N ;
   finpour
   retourner(R) ;
 fin
finfonction
```

Question 6.

1. Quels sont les avantages et les inconvénients de l'utilisation des listes par rapport à l'utilisation de tableaux pour implémenter une séquence de nombres ?
   Avantages : Il n'est pas nécessaire de connaître la taille maximum de la séquence de nombres.
   Inconvénients : tout nombre sera rangé avec au minimum la clé (pointeur) de son suivant par suite on utilise plus d'espace mémoire.

2. Soit L une liste simplement 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(4,L) ;
P=premier(L) ;
insererAprès (3,L,P) ;
insererAprès(4,L,P) ;
P=trouverDernier(L) ;
insérerAprès(5,L,P) ;
P=premier(L) ;
supprimerAprès(L,P) ;
P=suivant(L,premier(L)) ;
```

   P= adresse de 3…….
   Contenu(P)=3..…….
   L= (4,3,5)…….

   - Ecrire une fonction *comptePicListe* dont le résultat est le nombre de fois où un entier est égal à l'entier qui le précède plus un dans la liste L. On s'inspirera de la question 4.

```
fonction comptePicListe(ref B :tableau[1..N]d'entiers) :entier ;
  var c :entiers ;
  var p :^entier ;
 fonction picListe(ref L :listeSC d'entiers ; val P :^entier) :booléen ;
    début
       retourner(contenu(P)= =contenu(suivant(L,P))-1)
    fin
 finfonction
début
  c=0 ;
 si non(listeVide(L)) alors
    p=premier(L) ;
    tantque suivant(L,p) !=NIL faire
       si picListe(L,p) alors c=c+1 finsi;
      p=suivant(L,p) ;
    fintantque
 finsi
 retourner(c) ;
fin
finfonction
```

FIN