NOM: **DEUG A**

Module M2
janvier 1996

Examen d'Informatique

Avertissement important

1. *Le document suivant sert à la fois de sujet et de formulaire pour répondre aux questions. Il est impératif de l'utiliser pour composer l'examen.*
2. *Les questions suivantes sont indépendantes. Elles peuvent être traitées dans un ordre quelconque.*

On considère dans ce problème des tableaux de taille n qui contiennent une permutation des entiers $1, 2, \ldots, n$ c'est à dire tels que chaque entier $1, 2, \ldots, n$ apparait une fois et une seule.

Pour $n = 4$ on pourra par exemple avoir le tableau

$$a = \boxed{3 \mid 1 \mid 4 \mid 2}$$

Question 1

(3 points)

Ecrire en Pascal une fonction `permutation(a:Tableau)` à valeurs de type booléen qui prend la valeur vrai si un tableau de n entiers contient une permutation.

(on pourra par exemple compter le nombre de fois que chaque entier $i = 1, 2, \ldots, n$ apparait dans le tableau a)

```
function permutation (a: Tableau): boolean;
    var i: integer;
        test: Tableau;
        b: boolean;
    begin
        for i := 1 to N do test[i] := 0;
        for i := 1 to N do
            test[a[i]] := test[a[i]] + 1;
        b := true;
        for i := 1 to N do
            b := b and (test[i] = 1);
        permutation := b;
    end;
```

Question 2

(3 points)

On rappelle que si le tableau a contient une permutation, le tableau b contenant la permutation inverse est défini par

$$a[b[i]] = b[a[i]] = i$$

Par exemple, pour

$$a = \boxed{3}\boxed{1}\boxed{4}\boxed{2}$$

on a

$$b = \boxed{2}\boxed{4}\boxed{1}\boxed{3}$$

Ecrire une procedure prenant en paramètres deux tableaux a, b qui écrit dans le tableau b l'inverse de la permutation contenue dans a.

```
procedure inverse (a: Tableau; var b: Tableau);
    var i: integer;
    begin
       for i := 1 to N do
           b[a[i]] := i;
    end;
```

Question 3

(2 points)

Si les tableaux a et b contiennent des permutations, le tableau c contenant la permutation composée est défini par

$$c[i] = b[a[i]]$$

Par exemple, si

$$a = \boxed{3}\boxed{1}\boxed{4}\boxed{2}, \quad b = \boxed{4}\boxed{1}\boxed{2}\boxed{3}$$

alors

$$c = \boxed{2}\boxed{4}\boxed{3}\boxed{1}$$

Ecrire une procedure `composition(a,b,c)` qui calcule c.

```
procedure composition (a, b: Tableau; var c: Tableau);
    var i: integer;
    begin
       for i := 1 to N do
           c[i] := b[a[i]]
    end;
```

Question 4

(1 point)

On dit qu'un indice i est un *point fixe* d'une permutation contenue dans un tableau a si $a[i] = i$. Ecrire une fonction `fixes(a)` qui rend une valeur égale au nombre de points fixes de a.

```
function fixes (a: Tableau): integer;
    var i, val: integer;
    begin
        val := 0;
        for i := 1 to N do
            if a[i] = i then
                val := val + 1;
        fixes := val;
    end;
```

Question 5

(2 points)

Une *inversion* d'une permutation est une paire (i, j) d'indices avec $1 \leq i < j \leq n$ telle que $a[i] > a[j]$. On dit que la permutation est *paire* si le nombre de ses inversions est pair et *impaire* sinon. Ecrire une fonction `parite(a)` qui rend comme valeur $+1$ si la permutation contenue dans a est paire et -1 si elle est impaire.

```
function parite (a: Tableau): integer;
    var i, j, par: integer;
    begin
        par := 1;
        for i := 1 to n do
            for j := i + 1 to n do
                if a[i] ¿ a[j] then
                    par := − par;
        parite := par;
    end;
```

Question 6

(3 points)

La *table des inversions* d'une permutation $a[1], a[2], \ldots, a[n]$ est définie par: $inv[i]$ = le nombre d'éléments $< i$ à droite de i. On a par exemple

a	=	3	2	5	1	4
inv	=	0	1	2	0	2

Ecrire en Pascal une procédure qui calcule la table des inversions d'une permutation.

```
procedure TableInv (a: Tableau; var inv: Tableau);
    var i, j: integer;
    begin
        for i := 1 to N do begin
            inv[a[i]] := 0;
            for j := i + 1 to N do
                if a[i] > a[j] then
                    inv[a[i]] := inv[a[i]] + 1;
        end;
    end;
```

Question 7

(3 points)

La table des inversions permet de retrouver la permutation a. En effet, si $inv[n] = k$, la place de n est égale à $n - k$ puisqu'il a k éléments à sa droite. Ensuite la place de $n-1$ se calcule en partant de n et en laissant $inv[n-1]$ cases vides à gauche. Puis on calcule la place de $n-2$ et ainsi de suite. Par exemple, à partir de

$$inv = \boxed{0}\boxed{1}\boxed{2}\boxed{0}\boxed{2}$$

on obtient successivement

$$\boxed{-}\boxed{-}\boxed{5}\boxed{-}\boxed{-} \quad \boxed{-}\boxed{-}\boxed{5}\boxed{-}\boxed{4} \quad \boxed{3}\boxed{-}\boxed{5}\boxed{-}\boxed{4}\cdots$$

Ecrire la procedure qui calcule a à partir de inv.

```
procedure Inverse1 (inv: Tableau; var a: Tableau);
    var i, j, k: integer;
    begin
        for i := 1 to N do a[i] := 0;
        for i := N downto 1 do begin
            k := inv[i];
            j := N;
            while (k <> 0) do begin
                {on recule de k places vides}
                 if a[j] = 0 then
                    k := k - 1;
                j := j - 1;
            end;
            while a[j] <> 0 do j := j - 1;
            a[j] := i
            end;
    end;
```

Question 8

(3 points)

Une autre méthode pour retrouver la permutation a à partir de la table des inversions consiste à insérer les éléments en ordre croissant.

On place d'abord 1 en $a[1]$. On place ensuite 2 devant ou derrière 1 suivant que $inv[2] = 1$ ou 0, puis on place 3 et ainsi de suite.

Pour l'exemple précédent, avec

$$inv = \boxed{0 \mid 1 \mid 2 \mid 0 \mid 2}$$

on aura successivement

1				

2	1			

3	2	1		

3	2	1	4	

3	2	5	1	4

Ecrire la procedure avec cette méthode.

```
procedure Inverse2 (inv: Tableau; var a: Tableau);
    var i, k, j: integer;
    begin
        for i := 1 to N do begin {placer i}
            k := i - inv[i]; {k = place de i}
            for j := i downto k + 1 do
                a[j] := a[j - 1]; {décaler vers la droite}
            a[k] := i
        end;
    end;
```