EXERCICES SUR LES NOMBRES PREMIERS

EXERCICE1

Nombres de Mersenne:

a: Montrez que pour tout n entier naturel > 2 , si $2^n - 1$ est premier alors n est premier

b: Montrez que $2^{11} - 1$ n'est pas premier

EXERCICE2

Le petit Théorème de FERMAT:

Soit p un entier naturel premier et n un entier strictement compris entre 0 et p.

a : Montrez que C_p^n est divisible par p.

b : Montrez que, pour tout entier naturel a, $(a + 1)^p - a^p - 1$ est divisible par p.

c : Montrez que, pour tout b entier naturel, si $(b^p - b)$ est divisible par p alors $(b +1)^p - (b +1)$ l'est aussi.

d : Déduisez-en le petit théorème de Fermat:

"Pour tout entier p premier et tout a entier , $a^p \equiv a$ [p] "

e: Montrez que p est premier si et seulement si pour tout $r \in \{1 ; 2 ; ... ; p - 1\}$ on a $r^{p-1} \equiv 1$ [p]

EXERCICE3

Soit p un entier naturel premier. On note Ep l'ensemble {1 ; 2 ; .... ; p-1}.

a : Montrez que tout élément de Ep est premier avec p.

b : Montrez que pour tout a de Ep , il existe b unique dans Ep tel que $ab \equiv 1$ [p].

c : Déterminez les a éléments de Ep tels que $a^2 = 1$ [p].

d : Montrez que $1x2x3...x(p-1) \equiv (p -1)$ [p]

e : Déduisez-en que pour tout p entier naturel premier, $(p -1)! + 1$ est divisible par p.

(Ce résultat ainsi que sa réciproque est le théorème de Wilson

EXERCICE4

Montrez que si un entier naturel a exactement trois diviseurs dans **N** alors cet entier est le carré d'un nombre premier

EXERCICE5

a: Montrez que pour tout couple d'entier relatifs (x , y) , si $x^2 + y^2$ est divisible par 7 alors x et y sont aussi divisibles par 7.

b: Montrer que le seul triplet d'entiers naturels (x, y, z) vérifiant $x^2 + y^2 = 7z^2$

est (0 , 0 , 0)

EXERCICE6

p est un nombre premier > 2. On suppose qu'il existe a et b dans **N** tels que $p = a^2 + b^2$.

a: Déterminez les valeurs $p < 100$ possibles.

b: Montrez que pour tout x dans **N**, x^2 est congru à 0 ou 1 modulo 4.

c: Montrez alors que p est congru à 1 modulo 4.

d: Peut-écrire 2003 comme somme de deux carrés dans **N**?

EXERCICE7

p est un nombre premier. On note E l'ensemble {0 ; 1 ; 2 ; ... ; p - 1}et E^* l'ensemble E privé de 0.

a: Montrez que pour tout $a \in E^*$, il existe $b \in E^*$ unique tel que $ab \equiv 1$ [p].

b: Montrez que pour tout (a , b) dans E^2, on a : $ab \equiv 0$ [p] $\Rightarrow$ ($a = 0$ ou $b = 0$).

c: Montrez que pour tout X appartenant à E*, il existe au plus deux éléments dans E* vérifiant $x^2 \equiv X$ [p]

d: Déterminez les $x \in E^*$ tels que $x^2 \equiv 1$ [p].

On pose $p = 13$.

f: Quelles sont les valeurs possibles X appartenant à E telles qu'il existe x dans E tel que $x^2 \equiv X$ [p]?

g: Déterminez l'ensembles de x dans **Z** vérifiant l'équation $x^2 + 2x + 3 \equiv 0$ [p].

h: Déterminez l'ensemble des x dans **Z** tels que $x^2 + 5x - 6$ soit divisible par 13

EXERCICE8

En reprenant l'exercice 8: , résolvez les équations suivantes:

a: $x^2 + 5x \equiv 0$ [5]

b: $x - 5x + 2 \equiv 0$ [7]

c: $2x^2 + 4x + 1 \equiv 0$ [7]

d: $(x^2 - 1)^2 \equiv 9$ [11]

EXERCICE9

Deux nombres premiers n et m sont dits "jumeaux" si $n + 2 = m$.
Par exemple , les couples (11 , 13) , (17 , 19) , (41 , 43) sont des couples de nombres premiers jumeaux.
On considère un entier $n > 3$.

a. Montrez que si $(n , n + 2)$ est un couple de nombres premiers jumeaux alors n doit être congru à 2 modulo 3, autrement dit, on doit avoir , $n \equiv 2$ [3].
b. Montrez que si $(n , n +2)$ est un couple de nombres premiers jumeaux alors $n+ 4$ ne peut pas être premier.
c. Montrez que $(n , n +2)$ est un couple de nombres premiers jumeaux si et seulement si $n^2 + 2n$ a exactement 4 diviseurs dans **N**

EXERCICE10

Montrez que, pour tout b entier ≥ 3 , le nombre $x = 1 + b + 2b^2 + b^3 + b^4$ n'est pas un nombre premier

EXERCICE11

Le but de cet exercice est de déterminer le nombre de 0 que comporte $n!$ à la fin de son écriture en base 10.
Par exemple, 5! = 120 et a donc 1 "0" à la fin de son écriture en base 10.
a:
Montrez que si la décomposition d'un entier a en facteurs premiers est:
$a = 2^{a_1} \times 5^{a_2} \times 7^{a_3} \times \cdots$
alors a finit par exactement a_2 ou a_1 "0" en base 10.

Soit n un entier > 0. On pose Q_0 le quotient de la division euclidienne de n par 5.
b:
Montrez que : $n! = 5^{Q_0} . (Q_0!) . K_0$ où K_0 est un entier premier avec 5.
c: On pose Q_1 le quotient de la division euclidienne de Q_0 par 5.
Montrez que $n ! = 5^{Q_0 + Q_1} . (Q_1!) . K_1$ où K_1 est premier avec 5.
Expliquez pourquoi le nombre de 0 de n en base 10 est $Q_0 + Q_1$ + nombre de "0" de $(Q_1!)$
d: On définit alors la suite (Q_k) par:
Q_0 = quotient de la division euclidienne de n par 5 et pour tout $k \geq 0$,
Q_{k+1} = quotient de la division euclidienne de Q_k par 5.

Montrez qu'il existe un entier N tel que pour tout $k \geq N$, $Q_k = 0$.
Montrez que le nombre de 0 finissant l'écriture de $n!$ en base 10 est :
$Q_0 + Q_1 + Q_2 + ... + Q_N$

e: Application:
On pose $n = 100$.
Calculez Q_0 , Q_1 et Q_2.
Vérifez que 100! finit avec 24 "0" en base 10.

f: En utilisant la fonction Trunc ou Round de votre machine, déterminez le nombre de zéros finissant l'écriture de (123456789!) en base 10.
(*solution* : 30 864 192)

CORRECTION

EXERCICE1

a: Faisons un raisonnement par l'absurde. Supposons que n ne soit pas premier.
On a donc n = ab avec a et b entiers > 1.
Rappelons l'identité : $X^k - 1 = (X-1)(X^{k-1} + X^{k-2} + ... + X + 1)$.
On peut alors écrire:
$2^{ab} - 1 = (2^a)^b - 1 = (2^a - 1)[(2^a)^{b-1} + (2^a)^{b-2} + ... + 1]$, produit de deux entiers > 1.
D'où 2ab-1 n'est pas premier d'où la conclusion

b: $2^{11} - 1 = 2047 = 23x89$ donc $2^{11}-1$ n'est pas premier

EXERCICE2

a: Si p est premier et si n est entier avec 0 < n < p , alors:

$$C_p^n = \frac{p!}{n!(p-n)!} = \frac{p}{n}\frac{(p-1)!}{(n-1)![(p-1)-(n-1)]!} = \frac{p}{n}C_{p-1}^{n-1}$$

On a donc la relation :

$$nC_p^n = pC_{p-1}^{n-1}.$$

D'après le théorème de Gauss, comme n et p (car 1 < n < p et p premier), on peut dire que C_p^n est divisible par p.

b: Il suffit alors , pour voir que $(a + 1)^p - a^p - 1$ d'utiliser la formule du binôme de Newton, et de constater qu'en développant $(a + 1)^p - a^p - 1$, il ne reste que des termes divisibles par p, d'après la question précédente.

c: Même principe que la question précédente mais en faisant une récurrence sur b.

d: Conséquence directe des questions c: et b:

e: p est premier si et seulement si p est premier avec tout entier r appartenant à {1;2;...;p-1}.
Si p est premier alors pour tout r dans {1;2;...;p-1}, on a (r^p - r) divisible par p.
Or, (r^p - r) = r(r^{p-1} - 1).
Comme p et r sont premiers entre eux , on a alors (r^{p-1} - 1) divisible par p, ou encore,
$r^{p-1} \equiv 1\ [p]$

EXERCICE3

a: Evident car tout si a et p , avec a dans Ep ont un diviseur commun d $\geq$ 1 alors d est inférieur à a donc stirctement inférieur à p et comme p est premier, la seule valeur possible pour d est 1.

b: Si a est dans Ep, comme a et p sont premiers entre eux, on sait d'après le théorème de Bachet-Bezout, qu'il existe deux entiers naturels u et v tels que au+pv=1.
Soit u = Qp + b la division euclidienne de u par p. On a b dans {1;2;...;p-1}.
Effectivement, si b = 0 alors au + pv est divisble par p , ce qui contredit l'égalité au+pv=1.
Alors au + pv = a(Qp+b) + pv
= ab + (aQ+v)p
= 1
On a donc ab $\equiv$ 1 [p]. L'existence de b est donc assurée.
Pour l'unicité, supposons qu'il existe un autre entier c dans Ep tel que ac $\equiv$ 1 [p]
Alors a(b-c) est divisble par p. Comme a est premier avec p, on a donc (b-c) divisible par p.
Or, (b-c) est compris entre -(p-1) et (p-1) donc il ne peut pas être divisible par p.
D'où l'unicité de b.

c: $a^2 \equiv 1$ [p] si et seulement si (a-1)(a+1) est divisible par p.
a = 1 et a = (p-1) sont deux solutions évidentes.
Si a est dans {2;3;...;p-2} alors (a-1) et (a+1) sont dans {1;2;...;p-1}, donc premiers avec p.
Dans ce cas (a-1)(a+1) ne pas être divisible par p (car p premier).
Les seules solutions sont donc 1 et (p-1).

d: Pour p = 2,le résultat est évident car dans ce cas (p-1)! = 1! = 1 = (p-1) [p].
Pour p > 2 et premier:
Pour k compris strictement entre 1 et (p-1), il existe un k' unique distinct de k compris strictement entre 1 et (p-1) tel que kk' $\equiv$ 1 [p].

Dans le produit 1*2*3*...*(p-2)*(p-1), on regroupe alors les facteurs compris entre 2 et (p-2) deux par deux tels que le produit de ces facteurs soit identique à 1.
On a donc 1x(aa')x(bb')x(cc')x.....(dd')x(p-1) = 1x2x3x...x(p-1).
ce qui s'écrit 1x(p-1) $\equiv$ 1x2x3x...x(p-1) [p] d'où 1x2x3x...x(p-1) $\equiv$ (p-1) [p].

e: Comme (p-1) $\equiv$ -1 [p], on en déduit que 1x2x3x...x(p-1) +1 $\equiv$ 0 [p]
ou encore (p-1)! + 1 $\equiv$ 0 [p], c'est à dire (p-1)! + 1 est divisible par p.

EXERCICE4

Si n est le carré d'un nombre premier p, $n = p^2$, alors les diviseurs de n dans **N** sont 1 , p et p^2.
n a donc exactement 3 diviseurs dans **N**.

Réciproquement, si n a exactement 3 diviseurs dans **N**, comme 1 et n sont des diviseurs de n, n a un autre diviseur p compris strictement entre 1 et n.
p est premier, car si d divise p alors d divise n. Donc, d = 1 ou p car les seuls diviseurs de n < n sont 1 et p.
Donc, en particulier, p est le seul diviseur premier de n.
Donc, la décomposition de n en facteurs premiers est : $n = p^a$, avec a > 1.
Le nombre de diviseurs de n est alors (a+1), d'où a = 2. D'où $n = p^2$.

D'où la conclusion...

EXERCICE5

Montrez que pour tout couple d'entier relatifs (x , y) ,
si $x^2 + y^2$ est divisible par 7 alors x et y sont aussi divisibles par 7

Passons aux congruences modulo 7....

Pour un entier relatif a quelconque, on a

- **a = 0 ou a = 1 ou a = 2 ou a = 3 ou a = 4 ou a = 5 ou a = 6 modulo 7.**
  Ce sont simplement les restes possibles dans la division euclidienne de a par 7.
  Donc, sachant que si $a = b$ modulo 7 alors $a^2 = b^2$ modulo 7, les carrés modulo 7 sont:

- **0 = 0^2 ou 1 = 1^2 = 6^2 ou 2 = 3^2 = 4^2 ou 4 = 2^2 = 5^2 modulo 7.**
  On remarque alors que la seule possibilité d'avoir $x^2 + y^2 = 0$ modulo 7 est de

choisir
$x = 0$ et $y = 0$ modulo 7 , c.a.d, x et y divisibles par 7

EXERCICE6

a) Faites la liste .....

b)
Pour tout x dans **Z**, on a x congru à 0 ou 1 ou 2 ou 3 modulo 4.
Donc, x^2 est congru à 0^2 ou 1^2 ou 2^2 ou 3^2 modulo 4.
D'où x^2 est congru à 0 ou 1 modulo 4.

c)
Comme p est un nombre premier > 2, p est impair donc congru à 1 ou 3 modulo 4.
$p = a^2 + b^2$. Or, a^2 et $b^2 = 0$ ou 1 modulo 4.
Donc, modulo 4, les valeurs possibles de $a^2 + b^2$ sont 0 ou 1 ou 2.
Comme p est congru à 1 ou 3 modulo 4, on a alors p congru à 1 modulo4.

d) 2003 est premier (faites-vous la vérification!)
De plus, 2003 = 3 modulo 4.
Donc, d'après la question précédente, 2003 ne peut s'écrire sous la forme a^2+b^2 avec a et b entiers

EXERCICE9

a: n étant un entier premier > 3, n n'est pas divisible par 3 , donc : $n \equiv 1$ ou 2 [3]
Si de plus n+2 est aussi premier, on a alors $n+2 \equiv 1$ ou 2 [3]
Mais si $n \equiv 1$ [3] alors $n+2 \equiv 3$ [3] c.a.d , $n+2 \equiv 0$ [3]. Ce qui est impossible.
Donc, on doit avoir : $n \equiv 2$ [3]

b: Toujours suivant le même principe, et d'après la questin a:, si (n , n+2) est un couple de nombres premiers jumeaux,
alors $n \equiv 2$ [3] donc $n + 4 \equiv 6$ [3] , d'où $n + 4 \equiv 0$ [3].
Donc, $n + 4$ est divisible par 3 et > 3 , donc $n + 4$ n'est pas premier.

c: Par définition des nombres premiers, n est premiers si et seulement si n admet exactement 2 diviseurs dans **N**.
1 et n lui-même.
Or, $n^2 + 2n = n(n + 2)$.
Si n et $n + 2$ sont premiers alors $n(n + 2)$ est la décomposition de $n^2 + 2n$ en facteurs premiers.
donc $n^2 + 2n$ a bien 4 diviseurs : 1 , n , $(n + 2)$ et $n(n + 2)$.

Réciproquement.

Si $n^2 + 2n$ a exactement 4 diviseurs, comme 1 , n , n+2 et $n^2 + 2n$ sont des diviseurs de $n^2 + 2$n, on a là tous les

divisieurs de $n^2 + 2n$.

n+2 et $n^2 + 2n$ ne divise pas n. Donc, n n'admet aucun autre diviseur à part n et 1.

(Sinon, un tel diviseur p diviserait aussi $n^2 + 2n$, et $n^2 + 2n$ aurait plus de 4 diviseurs!)

Donc, n est premier.

Comme $n > 3$ et premier, n ne divise pas n+2.

Donc, pour les mêmes raisons, $(n + 2)$ n'admet pas d'autre diviseur à part 1 et $(n + 2)$.

Donc, $(n + 2)$ est aussi premier.

Conclusion:

n et $(n + 2)$ sont premiers si et seulement si $n^2 + 2n$ admet exactement 4 diviseurs

EXERCICE10

Montrez que, pour tout b entier ≥ 3 , le nombre $x = 1 + b + 2b^2 + b^3 + b^4$ n'est pas un nombre premier.

Remarquez simplement que $\mathbf{1 + b + 2b^2 + b^3 + b^4 = (1 + b^2)(1 + b + b^2)}$

MATHS ET INFORMATIQUE

Exercice 1

Écrire un programme permettant de résoudre le système de 2 équations à 2 inconnues :

$$\{u_1 x + v_1 y = w_1 u_2 x + v_2 y = w_2$$

On pourra imprimer les solutions à l'aide de l'instruction :

```
PRINT *, 'X = ', X, ', Y = ', Y
```

Exercice 2

Écrire un programme permettant de calculer les racines du trinôme du 2nd degré : $ax^2 + bx + c$. On s'assurera que a est non nul. Les racines, si elles existent, pourront être imprimées à l'aide de l'instruction :

```
PRINT *, 'X1 = ', X1, ', X2 = ', X2
```

Exercice 3

Écrire un programme calculant le nombre d'Or. Celui-ci peut être obtenu à partir de la suite de Fibonnacci u_n définie par :

$u_0 = 1$
$u_1 = 1$
...
$u_{n+1} = u_n + u_{n-1}$

La suite (u_{n+1} / u_n) converge vers le nombre d'Or.

Exercice 4

Écrire un programme permettant de déterminer les nombres premiers dans l'intervalle **[1,n]** à l'aide du crible d'Ératosthène. Il consiste à former une table avec tous les entiers naturels compris entre **2** et **n** et à rayer (mise à zéro), les uns après les autres, les entiers qui ne sont pas premiers de la manière suivante : dès que l'on trouve un entier qui n'a pas encore été rayé, il est déclaré premier, et on raye tous les multiples de celui-ci.

À la fin du procédé, les nombres non barrés sont des nombres premiers.

On tiendra compte du fait qu'un nombre donné peut déjà avoir été éliminé en tant que multiple de nombres précédents déjà testés.

Par ailleurs, on sait que l'on peut réduire la recherche aux nombres de **2** à ***n*** (si un entier non premier est strictement supérieur à ***n*** alors il a au moins un diviseur inférieur à *n* et aura donc déjà été rayé).

Exercice 5

Écrire un programme permettant de trier un vecteur de nombres en ordre croissant puis décroissant. On s'appuiera sur l'algorithme appelé *tri à bulle* qui consiste à comparer 2 éléments consécutifs et à les intervertir si nécessaire.

Si après avoir terminé l'exploration du tableau au moins une interversion a été effectuée, on renouvelle l'exploration, sinon le tri est terminé.

Exercice 6

Écrire un programme permettant d'effectuer le produit de 2 matrices `A` et `B`. Leurs profils seront définis à l'aide de constantes symboliques. La matrice résultat `C`sera imprimée à l'écran ligne par ligne avec l'instruction `PRINT` puis stockée dans un fichier binaire que l'on nommera « **`exo6.matrice`** ».

Exercice 7

Le fichier texte séquentiel « **`musiciens`** » est constitué de plusieurs enregistrements, chacun contenant un nom de musicien suivi de ses années de naissance et de mort.

Écrire un programme dont le but est de lire le fichier « **`musiciens`** » et de stocker les

enregistrements lus dans un fichier binaire à accès direct que l'on nommera « **musiciens.bin** ».

Exercice 8

Imprimer l'enregistrement du fichier « **musiciens** » dont le rang est entré au clavier. Son extraction sera effectuée à partir d'un fichier temporaire à accès direct, image du précédent.

On permettra la saisie de plusieurs rangs.

Exercice 9

Les enregistrements des fichiers séquentiels « **index_naissance.dat** » et « **index_deces.dat** » sont constitués d'une date de naissance (ou de décès) d'un musicien suivi de son rang dans le fichier « **musiciens.bin** » créé à l'exercice 7.

Écrire un programme permettant d'imprimer le ou les musiciens dont la date de naissance ou de mort est saisie au clavier. Le type de date désirée sera préalablement déterminé.

La sélection des enregistrements répondant aux choix spécifiés, s'effectuera par l'intermédiaire du fichier d'index correspondant au type de date.

On offrira la possibilité d'effectuer plusieurs recherches.

Exercice 10

Le but de cet exercice est de transformer la matrice stockée dans le fichier binaire « **exo6.matrice** ». Cette transformation consiste à modifier chaque élément à l'aide d'une fonction paramétrable de la forme $y = f(x)$.

On définira plusieurs fonctions de ce type. La valeur d'un entier lu dans une *namelist* indiquera la fonction à transmettre en argument de la procédure chargée d'effectuer la transformation.

Exercice 11

Trier les vecteurs lignes puis les vecteurs colonnes d'une matrice en utilisant l'algorithme *tri à bulle* et la matrice stockée dans le fichier binaire « **exo6.matrice** ».

On se définira une procédure effectuant le tri (croissant ou décroissant) des différents vecteurs au moyen d'une procédure interne.

Corrigé de l'exercice 1

```
program systeme
  implicit none
  real u1,u2
  real v1,v2
  real w1,w2
  real delta, delta_x, delta_y
  real x,y

              ! Valorisation des coefficients.
  u1 = 2; u2 = 4
  v1 = 5; v2 = 11
  w1 = 7; w2 = 6

            ! Calcul du déterminant principal.
  delta = u1*v2 - u2*v1
  if ( delta < 1e-6 ) then
    print *, "Le système n'a pas de solution unique."
    stop 4
  end if
              ! Calcul du déterminant en x.
  delta_x = w1*v2 - w2*v1
              ! Calcul du déterminant en y.
  delta_y = u1*w2 - u2*w1
                 ! calcul des solutions.
  x = delta_x/delta
  y = delta_y/delta
                ! Impression des solutions.
```

```
  print *, "x = ", x, ", y = ", y
end program systeme
```

Corrigé de l'exercice 2

```
program trinome
  implicit none
  real, parameter :: epsilon = 1e-6
  real a, b, c
  real delta, r_delta, x1, x2

             ! Valorisation des coefficients.
  a = 3.; b = 7.; c = -11.

                ! a doit être non nul.
  if ( a > -epsilon .and. a < epsilon ) &
    stop "a doit être non nul."

                ! calcul du déterminant.
  delta = b*b - 4*a*c
             ! cas du déterminant négatif.
  if( delta < -epsilon ) stop "Pas de racine réelle."

                ! cas du déterminant nul.
  if ( delta > -epsilon .and. delta < epsilon ) then
    x1 = -b/(2*a); x2 = x1
  else          ! cas du déterminant positif.
    r_delta = sqrt( delta )
    x1 = (-b - r_delta)/(2*a); x2 = (-b + r_delta)/(2*a)
  end if

                 ! Impression des racines.
  print *,"x1 = ", x1, ", x2 = ", x2
end program trinome
```

Corrigé de l'exercice 3

```
program nombre_dor
  implicit none
  real, parameter   :: epsilon = 1.e-5
  real              :: u_prec, u_cour
  real              :: v_prec, v_cour
  real              :: somme
  real              :: nombre_or

  nombre_or = (1. + sqrt(5.))/2.

  u_prec = 1.; u_cour = 1.
  do
    v_prec = u_cour/u_prec
    somme  = u_cour + u_prec
    u_prec = u_cour
    u_cour = somme
    v_cour = u_cour/u_prec
    if ( abs( (v_cour-v_prec)/v_prec ) < epsilon ) exit
  end do
```

```
  print*, "Limite de la suite (vn) : ", v_cour, &
          "Nombre d'or : ", nombre_or
end program nombre_dor
```

Corrigé de l'exercice 4

, firstnumber=1

```
program eratosthene
  implicit none
  integer, parameter    :: n = 1000
  integer, dimension(n) :: tab_nombres
  integer               :: imax
  integer i, j

  do i=2,n
    tab_nombres(i) = i
  end do

  imax = int(sqrt(real(n)))
  do i=2, imax
    if( tab_nombres(i) /= 0 ) then
      do j=i+1,n
        if ( tab_nombres(j) /= 0 .and. &
             mod( tab_nombres(j), i ) == 0 ) &
          tab_nombres(j) = 0
      end do
    end if
  end do

  print *,"Les nombres premiers entre 1 et ", n, " sont :"
  do i=2,n
    if ( tab_nombres(i) /= 0 ) print *,tab_nombres(i)
  end do
end program eratosthene
```

Corrigé de l'exercice 5 , firstnumber=1

```
program triabulle
  implicit none
  integer, parameter :: croissant=1, decroissant=2, n=10
  real, dimension(n) :: tab
  real               :: temp
  logical            :: tri_termine, expr1, expr2
  integer            :: sens, i

             ! Valorisation du vecteur
  data tab/0.76, 0.38, 0.42, 0.91, 0.25, &
           0.13, 0.52, 0.69, 0.76, 0.98/
  do sens=croissant, decroissant            ! Sens du tri
    do                                      ! Tri
     tri_termine = .true.
     do i=2,n
       expr1 = sens == croissant   .and. tab(i-1) > tab(i)
       expr2 = sens == decroissant .and. tab(i-1) < tab(i)
       if (expr1 .or. expr2) then
         tri_termine = .false.
         temp = tab(i-1); tab(i-1) = tab(i); tab(i) = temp
```

```
      end if
     end do
     if (tri_termine) exit
    end do
            ! Impression du vecteur trié
    if (sens == croissant)   print*, "Tri croissant "
    if (sens == decroissant) print*, "Tri décroissant "
    print*, tab
  end do
end program triabulle
```

Corrigé de l'exercice 6

```
program produit_matrice
  implicit none
  integer, parameter   :: n = 10, m = 5, p = 3
  real, dimension(n,m) :: a
  real, dimension(m,p) :: b
  real, dimension(n,p) :: c
  integer              :: i,j,k

          ! Valorisation des matrices A et B
  data a/0.00, 0.38, 0.42, 0.91, 0.25, &
         0.13, 0.52, 0.69, 0.76, 0.98, &
         0.76, 0.83, 0.59, 0.26, 0.72, &
         0.46, 0.03, 0.93, 0.05, 0.75, &
         0.53, 0.05, 0.85, 0.74, 0.65, &
         0.22, 0.53, 0.53, 0.33, 0.07, &
         0.05, 0.67, 0.09, 0.63, 0.63, &
         0.68, 0.01, 0.65, 0.76, 0.88, &
         0.68, 0.38, 0.42, 0.99, 0.27, &
         0.93, 0.07, 0.70 ,0.37, 0.44/

  data b/0.76, 0.16, 0.9047, &
         0.47, 0.48, 0.5045, &
         0.23, 0.89, 0.5163, &
         0.27, 0.90, 0.3190, &
         0.35, 0.06, 0.9866/
            ! Produit de matrice.
  do i=1,n
    do j=1,p
      c(i,j) = 0.
      do k=1,m
         c(i,j) = c(i,j) + a(i,k) * b(k,j)
      end do
    end do
  end do

           ! Impression de la matrice c.
  do i=1,n
    print*,c(i,:)
  end do

           ! Écriture de la matrice c dans un fichier.
  open( unit=1,           file="exo6.matrice", &
        status="replace", form="unformatted",  &
```

```
        action="write" )
  write( 1 ) c
  close( unit = 1)
end program produit_matrice
```

Corrigé de l'exercice 7 , firstnumber=1

```
program ecriture_musiciens
  character(len=80) :: mus
  integer           :: ios_mus
  integer           :: numrec

!  Ouverture du fichier des musiciens
!  ainsi que d'un fichier en écriture
!  à accès direct dans lequel on
!  va recopier le fichier précédent.

  open( unit=1,              file="musiciens",     &
        form="formatted",    status="old",         &
        action="read",       position="rewind" )

  open( unit=2,              file="musiciens.bin", &
        status="replace",                          &
        form="unformatted", access="direct",       &
        action="write",     recl=80 )

! On effectue la copie.
  numrec = 0
  read( unit=1, fmt='(a)', iostat=ios_mus ) mus
  do while ( ios_mus == 0 )
    numrec = numrec + 1
    write( unit=2, rec=numrec) mus
    read( unit=1, fmt='(a)', iostat=ios_mus ) mus
  end do
  close( unit=1 )
  close( unit=2 )
end program ecriture_musiciens
```

Corrigé de l'exercice 8

```
program musiciens
  implicit none
  character(len=80) :: mus
  integer           :: ios_mus, ios_stdin
  integer           :: numrec, rang

! Ouverture du fichier des musiciens
! ainsi que d'un fichier temporaire
! à accès direct dans lequel on
! va recopier le fichier précédent.
  open( unit=1,              file="musiciens",   &
        form="formatted",    status="old",       &
        action="read",       position="rewind" )
  open( unit=2,              status="scratch",   &
        form="formatted",    access="direct",    &
        action="readwrite", recl=80 )

! On effectue la copie.
  numrec = 0
  read( unit=1, fmt='(a)', iostat=ios_mus ) mus
  do while ( ios_mus == 0 )
    numrec = numrec + 1
    write( unit=2, rec=numrec, fmt='(a)' ) mus
    read( unit=1, fmt='(a)', iostat=ios_mus ) mus
  end do
```

```
  close( unit=1 )
! On demande un rang de musicien.

  print *,"Entrez le rang d'un musicien :"
  read( unit=*,  &
        fmt=*,   &
        iostat=ios_stdin ) rang
  do while ( ios_stdin == 0 )
    read( unit=2,    &
          rec=rang,  &
          fmt='(a)', &
          iostat=ios_mus ) mus
    if ( ios_mus /= 0 ) then
      print *,"Le musicien de rang ", &
                rang, "n'existe pas"
    else
      print '("musicien de rang",i3," ==> ", a)', &
             rang,trim(mus)
    end if
    print '(/,"Entrez le rang d''un musicien :")'
    read( unit=*, fmt=*, iostat=ios_stdin ) rang
  end do
  close( unit=2 )
end program musiciens
```

Corrigé de l'exercice 9 , firstnumber=1

```
program sequentiel_indexe
  implicit none
  character(len=19), dimension(2), parameter :: f_index = &
        (/ "index_naissance.dat", "index_deces.dat    " /)
  character(len=80) :: mus
  integer           :: numrec, ios_index
  integer           :: date_saisie, date_lue
  integer           :: critere
  logical           :: trouve

! Ouverture du fichier des musiciens à accès direct en lecture
! et des fichiers d'index.
  open ( unit=1,          file = f_index(1),                &
         status="old",    form="formatted", action="read" )
  open ( unit=2,          file = trim(f_index(2)),          &
         status="old",    form="formatted", action="read" )
  open ( unit=3,          file="musiciens.bin",             &
         status="old",    form="unformatted",               &
         access="direct", action="read", recl=80 )
  do
    print*,'-------------------------------'
    print*,'Choix du critère de recherche : '
    print*,'- par date de naissance (1)'
    print*,'- par date de décès     (2)'
    print*,'- QUITTER               (3)'
    read*, critere
    print*,'-------------------------------'

    trouve = .false.
    select case (critere)
      case(1) ! Recherche par date de naissance.
        print*, "Entrer une date de naissance d'un musicien"
        rewind( unit=critere )
      case(2) ! Recherche par date de décès.
        print*, "Entrer une date de décès d'un musicien"
        rewind( unit=critere )
```

```
      case default ! Quitter
        exit
    end select
    read *, date_saisie
! Recherche de la date saisie dans le fichier d'index.
    read( unit=critere, fmt=*,                &
          iostat=ios_index ) date_lue, numrec
    do while( ios_index == 0 )
      if ( date_lue == date_saisie ) then
! On lit l'enregistrement correspondant.
        trouve = .true.
        read( unit=3, rec=numrec ) mus
        print *,trim(mus)
      end if
      read( unit=critere, fmt=*,                &
            iostat=ios_index ) date_lue, numrec
    end do
    if ( .not. trouve ) &
      print *,"Aucun musicien ne répond au critère indiqué."
    print '(/)'
  end do
  close( unit=1 )
  close( unit=2 )
  close( unit=3 )
end program sequentiel_indexe
```

Corrigé de l'exercice 10 , firstnumber=1

```
program mat_transf
  implicit none
  integer, parameter   :: n = 10, m = 3
  real, dimension(n,m) :: mat
  integer              :: choix_methode, ios, num_ligne
  real, external       :: carre, identite, logarithme
  real, intrinsic      :: sqrt
  namelist/methode/choix_methode

        ! Ouverture du fichier contenant la matrice.
  open( unit=1,              file="exo6.matrice", &
        form="unformatted", action="read",       &
        status="old",       position="rewind",   &
        iostat=ios )
  if (ios /= 0) &
    stop 'Erreur à l''ouverture du fichier "exo6.matrice"'
          ! Lecture de la matrice.
  read(1) mat
  close(1)
         ! Ouverture du fichier contenant
              ! la namelist "methode".
  open( unit=1,           file="exo10.namelist", &
        form="formatted", action="read",         &
        status="old",     position="rewind",     &
        iostat=ios )
  if (ios /= 0) &
    stop 'Erreur à l''ouverture du fichier "exo10.namelist"'
  read( unit=1, nml=methode )
  close( unit=1 )
        ! Transformation de la matrice à l'aide
                ! de la méthode choisie.

  select case( choix_methode )
    case (1)
      call transform( mat, n, m, identite )
```

```
    case (2)
      call transform( mat, n, m, carre )
    case (3)
      call transform( mat, n, m, sqrt )
    case (4)
      call transform( mat, n, m, logarithme )
  end select

        ! Sauvegarde de la matrice transformée dans
                ! le fichier "exo6_matrice_transf".

  open( unit=1,           file="exo6_matrice_transf", &
        form="formatted", action="write",             &
        status="replace", iostat=ios )

  if ( ios /= 0 ) &
    stop "Erreur lors de l''ouverture &
         &du fichier ""exo6_matrice_trans"""

  do num_ligne=1,n
    write( unit=1, fmt='(3f10.6)' ) mat(num_ligne,:)
  end do
  close( unit=1 )
end program mat_transf
              ! Procédure de transformation.
subroutine transform( t, n, m, f )
  implicit none
  integer               :: n, m, i, j
  real, dimension(n,m)  :: t
  real                  :: f

  do i=1,n
    do j=1,m
      t(i,j) = f(t(i,j))
    end do
  end do
end subroutine transform
       ! Définitions des fonctions de transformation.
function identite(x)
  implicit none
  real x, identite
  identite = x
end function identite

function carre(x)
  implicit none
  real x, carre
  carre = x*x
end function carre

function logarithme(x)
  implicit none
  real x, logarithme
  logarithme = log(x)
end function logarithme
```

Corrigé de l'exercice 11

```
program tri_matrice
```

```
  implicit none
  integer, parameter   :: n=10, m=3
  real, dimension(n,m) :: mat
  integer              :: ios
  integer              :: i, j
           ! Lecture de la matrice à trier.
  open( unit=1,              &
        file="exo6.matrice", &
        form="unformatted",  &
        status="old",        &
        action="read",       &
        position="rewind",   &
        iostat=ios )
  if ( ios /= 0 ) stop "Erreur à l'ouverture du fichier &
                    &""exo6.matrice"""
  read( unit=1 ) mat; close( unit=1 )
  call tri( mat, n, m ) ! Tri de la matrice lue.
             ! Écriture de la matrice triée.
  open( unit=1,           file="exo11.matrice_triee", &
        form="formatted", status="replace",           &
        action="write",   position="rewind",          &
        iostat=ios )
  if ( ios /= 0 ) stop "Erreur à l'ouverture du fichier &
                    &""exo11.matrice_triee"""
  do i=1,n
    write( unit=1, fmt='(3F7.3)' ) mat(i,:)
  end do
  close( unit=1 )
end program tri_matrice
                  ! Procédure de tri.
subroutine tri( mat, n, m )
  implicit none
  integer              :: n, m
  real, dimension(n,m) :: mat
  integer              :: ligne, col

  do ligne=1,n      ! Tri des lignes.
    call tri_vecteur( mat(ligne,:), m )
  end do
  do col=1,m        ! Tri des colonnes.
    call tri_vecteur( mat(:,col), n )
  end do
  contains
          ! Procédure de tri d'un vecteur.
  subroutine tri_vecteur( v, n )
    integer              :: n, i
    real, dimension(n) :: v
    logical              :: tri_termine
    real                 :: temp
    do
     tri_termine = .true.
     do i=2,n
       if ( v(i) > v(i-1) ) then
         tri_termine = .false.
         temp = v(i-1); v(i-1) = v(i); v(i) = temp
       end if
     end do
     if (tri_termine) exit
    end do
  end subroutine tri_vecteur
end subroutine tri
```