

Correction TD 2
Algorithmique II SMI3

Exercice 1

1.

```
Variables N, i, PG en Entier
Debut
PG ← 0
Pour i ← 1 à 20
    Ecrire "Entrez un nombre : "
    Lire N
    Si i = 1 ou N > PG Alors
        PG ← N
    FinSi
i Suivant
Ecrire "Le nombre le plus grand était : ", PG
Fin
```

En ligne 3, on peut mettre n'importe quoi dans PG, il suffit que cette variable soit affectée pour que le premier passage en ligne 7 ne provoque pas d'erreur.

2.

Pour la version améliorée, cela donne :

```
Variables N, i, PG, IPG en Entier
Debut
PG ← 0
Pour i ← 1 à 20
    Ecrire "Entrez un nombre : "
    Lire N
    Si i = 1 ou N > PG Alors
        PG ← N
        IPG ← i
    FinSi
i Suivant
Ecrire "Le nombre le plus grand était : ", PG
Ecrire "Il a été saisi en position numéro ", IPG
Fin
```

3.

```
Variables N, i, PG, IPG en Entier
Debut
N ← 1
i ← 0
PG ← 0
TantQue N <> 0
    Ecrire "Entrez un nombre : "
    Lire N
    i ← i + 1

    Si i = 1 ou N > PG Alors
        PG ← N
        IPG ← i
    FinSi
FinTantQue
Ecrire "Le nombre le plus grand était : ", PG
Ecrire "Il a été saisi en position numéro ", IPG
Fin
```

Exercice 2 :

1- Algorithme MinMax ;

```
Var    T :Tableau[1..100] de entier ;
        I,N,Max,Min,S :entier ; Moy :reel ;

Debut
    /*lecture de la taille exacte
    Ecrire('Donner la taille du tableau N≤100') ;
    Repeter Lire(N) Jusqu'à N>0 et N≤100 ;
    /*Lecture des éléments de T
    Pour I←1 à N Faire Lire(T[I]) ; Fait ;
    /*Initialisation
    Min←T[1] ; Max←T[1] ; S←0 ;
    Pour I←1 à N Faire
        Si Max<T[I] Alors Max←T[I] Fsi ;
        Si Min>T[I] Alors Min←T[I] Fsi ;
        S←S+ T[I] ;

    Fait ;
    Moy←S/N ;
    Ecrire('Maximum=',Max,' Minimum=',Min,' Moyenne=',Moy) ;

Fin.
```

2- Algorithme Prod ;

```
Var    T :Tableau[1..100] de entier ;
        I,N,P,Nbp :entier ;

Debut
    /*lecture de la taille exacte
    Ecrire('Donner la taille du tableau N≤100') ;
    Repeter Lire(N) Jusqu'à N>0 et N≤100 ;
    /*Initialisation
    P←1 ; Nbp←0 ;
    /*Lecture des éléments de T et traitement en même temps
    Pour I←1 à N Faire
        Lire(T[I]) ;
        Si T[I]>0 Alors Nbp←Nbp+1 Fsi ;
        P←P* T[I] ;

    Fait ;
    Ecrire('Produit=',P,' Nb val positives=',Nbp) ;

Fin
```

3- Algorithme Prod ;

```
Var    T1,T2,T3 :Tableau[1..100] de entier ;
        I,N,PS :entier ;

Debut
    /*lecture de la taille exacte
    Ecrire('Donner la taille du tableau N≤100') ;
    Repeter Lire(N) Jusqu'à N>0 et N≤100 ;
    /*Lecture des éléments de T1 ensuite T2 ne pas lire dans la même boucle
    Pour I←1 à N Faire Lire(T1[I]) ; Fait ;
    Pour I←1 à N Faire Lire(T2[I]) ; Fait ;
    PS←0 ; /*initialiser Produit scalaire à 0
    /*La somme de T1 et T2 dans T3
    Pour I←1 à N Faire
        T3[I]←T1[I]+ T2[I];
        PS←PS+ T1[I]* T2[I];

    Fait ;
    Ecrire('Produit Scalaire=',PS) ;
    Ecrire('Somme des vecteurs') ;
    Pour I←1 à N Faire Ecrire (T3[I]) ; Fait ;

Fin.
```

4- **Algorithme** Position ;

Var T,Pos:Tableau[1..100] de entier ;
I,J,N,Val :entier ;

Debut

*/*lecture de la taille exacte*

Ecrire('Donner la taille du tableau $N \leq 100$ ') ;

Repeter Lire(N) **Jusqu'**à $N > 0$ et $N \leq 100$;

Pour I←1 à N **Faire** Lire(T[I]) ; **Fait** ;

Ecrire('Donner Val') ; Lire(Val) ;

*/*Recherche de val et sa position*

J←0 ;

Pour I←1 à N **Faire**

Si T[I]=Val **Alors** J←J+1 ;Pos[J]←I **Fsi** ;

Fait ;

Si J=0 **Alors** Ecrire(Val,'non trouvée')

Sinon Ecrire(Val,'trouvée aux positions :') ;

Pour I←1 à J **Faire** Ecrire (Pos[I]) ; **Fait** ;

Fsi ;

/ Si on initialise J à 1, son incrémentation se fait après l'affectation et la*

*/*dimension de Pos devient J-1*

Fin.

5- **Algorithme** Inverse ;

Var T :Tableau[1..100] de entier ;
I,J,X,N:entier ;

Debut

*/*lecture de la taille exacte*

Ecrire('Donner la taille du tableau $N \leq 100$ ') ;

Repeter Lire(N) **Jusqu'**à $N > 0$ et $N \leq 100$;

*/*Lecture des éléments de T*

Pour I←1 à N **Faire** Lire(T[I]) ; **Fait** ;

*/*Inverser*

I←1 ; J←N ;

Tantque I<J

Faire

 X←T[I] ; T[I]←T[J] ; T[J]←X;

 I←I+1 ; J←J-1;

Fait ;

*/*Affichage du nouveau tableau T*

Pour I←1 à N **Faire** Ecrire(T[I]) ; **Fait** ;

Fin.

```

6- Algorithme SupprimeZero ;
Var   T :Tableau[1..100] de entier ;
        I,J,N:entier ;

Debut
    /*lecture de la taille exacte
    Ecrire('Donner la taille du tableau N≤100') ;
    Repeter Lire(N) Jusqu'à N>0 et N≤100 ;
    /*Lecture des éléments de T
    Pour I←1 à N Faire Lire(T[I]) ; Fait ;
    /*la suppression des zéro revient à décaler les valeurs non nulles
    I←1 ;
    Tantque I≤N
    Faire
        Si T[I]=0
        Alors /*boucle de décalage
            Pour J←I à N-1
            Faire T[J]←T[J+1]; Fait ;
            N←N-1 /*changer la taille du tableau
        Fsi ;
        I←I+1 ;
    Fait ;
    /*Affichage du nouveau tableau T
    Pour I←1 à N Faire Ecrire(T[I]) ; Fait ;

Fin.

```

```

7- Algorithme NegPuisPos ;
Var   T :Tableau[1..100] de entier ;
        I,J,N,X:entier ;

Debut
    /*lecture de la taille exacte
    Ecrire('Donner la taille du tableau N≤100') ;
    Repeter Lire(N) Jusqu'à N>0 et N≤100 ;
    /*Lecture des éléments de T
    Pour I←1 à N Faire Lire(T[I]) ; Fait ;
    /*passer les valeurs négatives au début
    J←1 ;
    Tantque J≤N et T[J]<0 Faire J←J+1 ; Fait ;
    /*déplacer les valeurs négatives au début (trouver la première valeur positive I)
    I←J ;
    Tantque J≤N
    Faire
        Si T[J]<0
        Alors /*permuter
            X← T[I] ; T[I]←T[J]; T[J]←X;
            I←I+1
        Fsi ;
        J←J+1 ;
    Fait ;
    /*Affichage du nouveau tableau T
    Pour I←1 à N Faire Ecrire(T[I]) ; Fait ;

Fin.

```

Exercice 3 :

```

Var   T1,T2,T :Tableau[1..250] de entier ;
        I,J,K,N :entier ;
Debut
    /*lecture de la taille exacte
    Ecrire('Donner la taille du tableau  $N \leq 250$ ') ;
    Repeter Lire(N) Jusqu'à  $N > 0$  et  $N \leq 250$  ;
    Pour I←1 à N Faire Lire(T[I]) ; Fait ;
    J←1 ;K←1 ; /*compteurs pour les tableaux pairs et impairs
    Pour I←1 à N Faire
        Si T[I] MOD 2=0 Alors T2[J]← T[I] ; J←J+1
        Sinon T1[K]← T[I] ; K←K+1
    Fsi ;
    Fait ;

    Ecrire('Vecteur pairs') ; Pour I←1 à J Faire Ecrire(T2[I]) ; Fait ;
    Ecrire('Vecteur impairs') ; Pour I←1 à K Faire Ecrire (T1[I]) ; Fait ;

Fin.

```

Exercice 4 :

```

Variables Nb, Nbpos, Nbneg : entiers
Tableau T[100] :entier
Debut
Ecrire( "Entrez le nombre de valeurs : )
Lire( Nb)
Nbpos ← 0
Nbneg ← 0
Pour i ← 0 à Nb - 1
    Ecrire( "Entrez le nombre n° ", i )
    Lire (T[i])
    Si T[i]> 0 alors
        Nbpos ← Nbpos + 1
    Sinon
        Nbneg ← Nbneg + 1
Finsi
Ecrire ("Nombre de valeurs positives : ", Nbpos)
Ecrire ("Nombre de valeurs négatives : ", Nbneg)
Fin

```

Exercice 5 :

Algorithme Matrice ;

Var I,J,N,M,Val,Nbv :entier ; Existe :booleen ;

A : Tableau[1..20,1..30] de entier;

AT : Tableau[1..20,1..30] de entier;

Début

Repeter Lire(N) ; **Jusqu'à** (N>0) et (N≤20) ;

Repeter Lire(M) ; **Jusqu'à** (M>0) et (M≤30) ;

//Lecture de la Matrice

Pour I ←1 à N **Faire Pour** J ←1 à M **Faire**
Lire(A[I,J]) ;

Fait ;

Fait ;

Ecrire('Donner Val') ; Lire(Val) ;

*/*Recherche de Val*

Existe←Faux ; I←1 ;

Tanque I ≤N et Non Existe

Faire J←1 ;

Tanque J ≤M et Non Existe

Faire

Si A[I,J]=Val **Alors** Existe←Vrai **Fsi** ;

J←J+1 ;

Fait ;

I←I+1 ;

Fait ;

Si Existe **Alors** Ecrire(Val,'Existe') **Sinon** Ecrire(Val,'Non trouvée') **Fsi** ;

/ nombre de voyelles dans A*

Nbv←0 ;

Pour I ←1 à N **Faire Pour** J ←1 à M **Faire**

Cas A[I,J] **Vaut**

'a','e','i','o','u','y' : Nbv←Nbv+1 ;

Fincas ;

Fait ;

Fait ;

Ecrire('Nombre de voyelles est :',Nbv) ;

Fin.