

TP 2: Les fonctions et les procédures

Exercice 1. -déclaration et appel de deux fonctions-

- Qu'affiche l'algorithme suivant après son exécution?
- Quels sont les paramètres formels et les paramètres effectifs?

```

Algorithme exo1
Var a,b,c:réel

Fonction somme (x,y:réel):réel
Début
  somme ← x+y
FinFonction
Fonction sommeCar (x,y:réel):réel
Début
  sommeCar ← x*x+y*y
FinFonction

Début      {Algorithme principal}
  a ← 3
  b ← 4
  c ← somme(a,b)/2
  Ecrire("La somme de a et b = ",somme(a,b))
  Ecrire("La somme des carrés de a et b = ",sommeCar(a,b))
  Ecrire("c = ",c)
Fin

```

Exercice 2. -déclaration et appel d'une fonction-

Parmi les appels de la fonction essai numérotés dans l'algorithme suivant, certains ne sont pas corrects, expliquer pourquoi.

```

Algorithme exo2
Var a,b:réel
      c:entier
Fonction essai (x:réel,y:entier):réel
Début
  essai ← x/y
FinFonction

Début      {Algorithme principal}
  a ← 3.0
  b ← 2.0
  c ← 1
  Ecrire(essai(a,b))           (1)
  Ecrire(essai(a,c))           (2)
  essai(a,c)                   (3)
  a ← essai(a,c)               (4)
  a ← essai(essai(b,c),c)      (5)
  a ← essai(b,c)               (6)
  Ecrire(essai(1.5,2))         (7)
  Ecrire(essai(1.5,2.5))       (8)
  b ← essai((a/b)+2,c)         (9)
  b ← essai(a/b,c/b)          (10)
Fin

```

Exercice 3.

En utilisant une fonction, écrire un **algorithme** qui calcule et affiche le minimum de deux nombres entiers.

Exercice 4.

En utilisant une fonction, écrire un **programme Pascal** qui calcule et affiche la surface d'un trapèze.

Exercice 5.

Dans un **programme Pascal**:

- Écrire une fonction (*puissance*) qui calcule la puissance d'un nombre entier (x^n).
- Lire deux nombres entiers a et b , ensuite calculer et afficher a^b et b^a .

Exercice 6.

Dans un **programme Pascal**:

- Écrire une fonction (*pgcd*) qui calcule le PGCD de deux nombres entiers positifs.
- Calculer et afficher le PGCD de 114 et 36 et le PGCD de 230 et 60.

Exercice 7.

Dans un **algorithme principal**:

- Écrire une fonction (*fact*) qui calcule le factoriel d'un nombre entier positif.
- Lire deux valeurs entières positives Q et P, ensuite, calculer et afficher $(Q!(Q-P)!)/P!$

Exercice 8. -déclaration et appel d'une procédure et d'une fonction-

Quelles seront les valeurs des variables a, b, c et d après l'exécution de l'algorithme suivant?

```

Algorithme exo8
Var a,b,c,d:réel

Fonction somme (x,y:réel):réel
Début
  somme ← x+y
FinFonction

Procédure sommeCar (x,y:réel,var z:réel)
Début
  z ← x*x+y*y
FinProcédure

Début      { Algorithme principal }
  a ← 2
  b ← 4
  sommeCar (a,b,c)
  sommeCar (somme (a,1),b,d)
Fin

```

Exercice 9. - déclaration et appel d'une procédure-

Considérons l'algorithme ci-après.

1. Quels sont les appels qui ne sont pas corrects? Justifier vos réponses.
2. L'instruction suivante est elle juste ? Si oui, que fait-elle?

multiplier(4.0,3.0, c)

```

Algorithme exo9
VAR a,b,c: réel
        d: entier

Procédure multiplier (x,y: réel; var z: réel)
Début
    z ← x*y
    Ecrire(z)
FinProcédure

Début    {Algorithme principal}
    Lire (a,b)

    Ecrire(multiplier(a,b,c))    (1)
    multiplier(a,b,c)           (2)
    multiplier(a,b,a)           (3)
    c ← multiplier(a,b,c)       (4)
    multiplier(a,b)             (5)
    multiplier(a,b,d)           (6)

Fin

```

Exercice 10. - variables globales et variables locales-

```

Algorithme exo10
Var a,b,c: Réel

Procédure procl (x,y: Réel)
Var x1,y1: Réel
Début
    a ←x*y           ①
    Ecrire(x1)       ②
    b ←x*y2          ③
FinProc

Procédure proc2(z,t: Réel)
Var x2,y2: Réel
Début
    x2 ← a*b         ④
    y2 ← y1          ⑤
    Ecrire(y)        ⑥
FinProc

Début    {Algorithme principal}
    Lire (a,b)       ⑦
    Lire (x1,x2)     ⑧
    c ← z*x          ⑨

Fin

```

1. Citer tous les variables globales et tous les variables locales.
2. Parmi les instructions numérotées, lesquelles ceux qui ne sont pas autorisées ? Justifiez votre réponse.

Exercice 11. -passage des paramètres par valeur et par adresse (par variable)-

Soit le **programme Pascal** suivant, sachant que la procédure *echanger* permet d'échanger les valeurs de deux variables:

```

program exo11;
uses crt;
var a,b:integer;

procedure echanger(x,y:integer);
var z:integer;
begin
  z:=x;
  x:=y;
  y:=z;
end;

begin
  clrscr;
  a:=5;
  b:=10;
  writeln('Avant appel: a= ',a,' b= ',b);
  echanger(a,b);
  writeln('Après appel: a= ',a,' b= ',b);
  readkey;
end.

```

1. Taper et exécuter ce programme. Que remarquez-vous ?
2. Modifier l'entête de la procédure de la manière suivante et exécuter le programme à nouveau:

```
procedure echanger(var x:integer; y:integer);
```

Que remarquez-vous ?

3. Modifier l'entête de la procédure de la manière suivante et exécuter le programme à nouveau:

```
procedure echanger(var x:integer; var y:integer);
```

Que remarquez-vous ?

4. Les paramètres x et y sont-ils en entrée, en sortie ou en entrée/sortie ? Conclusion ?
5. Faire cet appel: `echanger(5,10)` et exécuter le programme à nouveau. Remarque ? Conclusion ?

Exercice 12:

Dans un **programme Pascal**:

1. Ecrire une fonction *premier* qui vérifie si un nombre entier positif est premier ou non.
2. Ecrire une procédure *nombre_pre* qui affiche tous les nombres premiers inférieurs à un nombre entier donné N . (Utiliser la fonction *premier*)
3. Afficher dans le programme principal tous les nombres premiers qui sont inférieurs à 1000.

Exercice 13: On définit un nouveau type nommé *mat* en **langage Pascal** comme suit:

```

Const N=3; M=3;
Type mat =array[1..N, 1..M] of integer;

```

1. Ecrire une procédure qui permet de lire (remplir) une matrice de type *mat*.
2. Ecrire une procédure qui permet d'afficher une matrice de type *mat*.
3. Ecrire une procédure qui calcule la somme de deux matrices de type *mat*.
4. Ecrire une procédure qui calcule le transposé d'une matrice de type *mat*.
5. Ecrire une procédure qui calcule le produit de deux matrices de type *mat*.
6. Calculer et afficher la somme des matrices X et Y.
7. Calculer et afficher le produit des matrices X et Y.
8. Calculer et afficher les transposés des matrices X et Y.
9. Calculer et afficher la somme des transposés des matrices X et Y.

$$X = \begin{pmatrix} 3 & -2 & 1 \\ 4 & 7 & 5 \\ 8 & 0 & -4 \end{pmatrix}$$

$$Y = \begin{pmatrix} 7 & 4 & 1 \\ 1 & -1 & 9 \\ 6 & 0 & 3 \end{pmatrix}$$

