



Ministère de l'enseignement supérieur et de la recherche scientifique  
Université Djillali BOUNAAMA - Khemis Miliana (UDBKM)  
Faculté des Sciences et de la Technologie  
Département de Mathématiques et d'Informatique



## Chapitre 1

# Les sous-programmes : Exercices corrigés

**MI-L1-UEF221 : Algorithmiques et Structures de Données II**

**Nouredine AZZOUZA**

n.azzouza@univ-dbk.m.dz

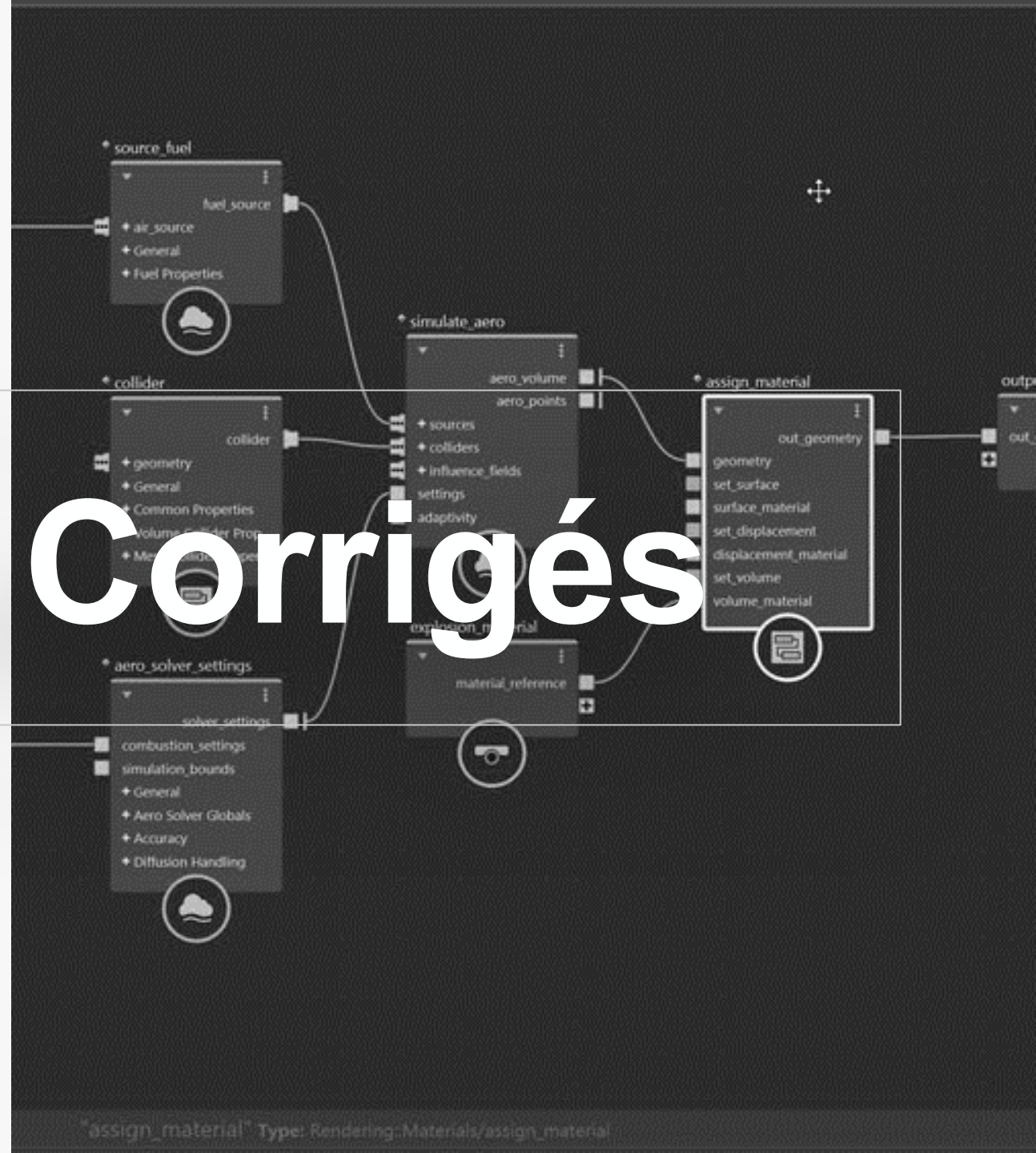
# Plan du Cours

**1. Exercices Corrigés**

**2. Exercices d'Examen**

**3. Exercices supplémentaires**

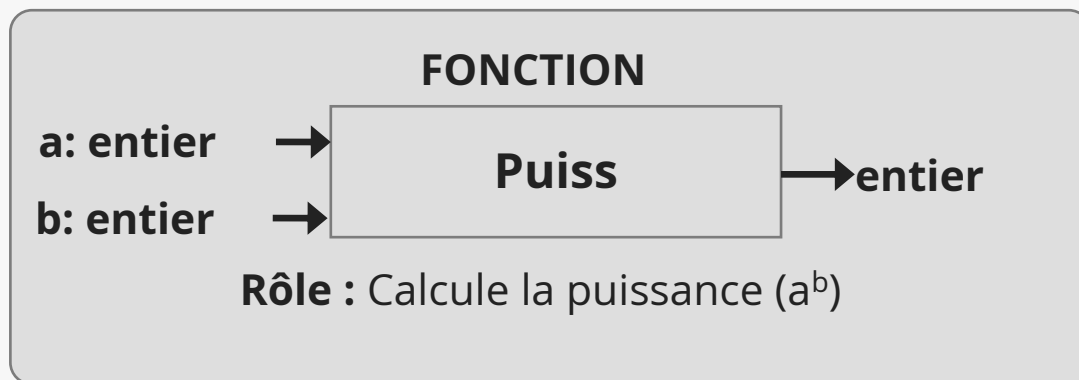
# = Exercices Corrigés



## Exercices 01

- ✓ Ecrire une fonction **puissance** qui calcule  $a^b = a \times a \times a \times \dots \times a$  ( $b$  fois);  $a$  et  $b$  étant des entiers positifs.

✓ **Solution :**



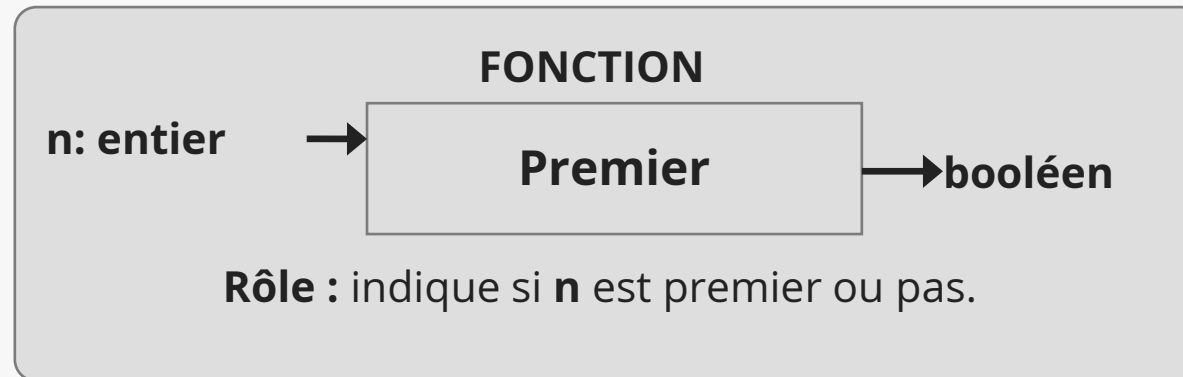
```
Fonction Puiss (a, b: entier): entier;  
Var P, i: entier;  
Début  
    P ← 1;  
    Pour i ← 1 à b Faire P ← P * a;  
  
    Puiss ← P;  
Fin;
```



## Exercices 02

- ✓ Sachant qu'un nombre premier est un nombre qui n'accepte aucun diviseur excepté 1 et lui-même.
  1. Ecrire une fonction **Premier** qui indique si un nombre entier N est premier ou pas.
  2. Construire l'algorithme qui nous donne les N premiers nombres premiers.

### ✓ Solution :



## Exercices 02 : Fonction Premier

```
Fonction Premier(n: entier): booléen;  
Var P, i: entier;  
        continue : booléen;  
Début  
    continue ← vrai;  
    i ← 2;  
    TQ (continue == vrai) et (i < n DIV 2) Faire  
        Dtq  
            Si n MOD i = 0 Alors continue ← faux;  
            Sinon i ← i+1;  
        Ftq  
    Si continue = vrai Alors Premier ← vrai;  
    Sinon Premier ← faux;  
Fin;
```



## Exercices 02 : Algorithme principal

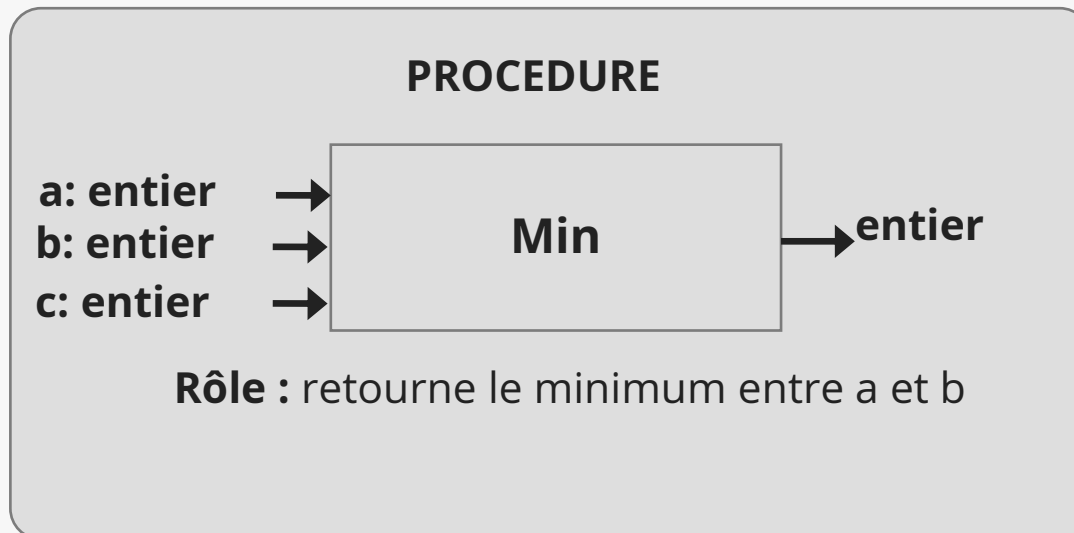
```
Algorithme exo_2;  
Var    i, cpt: entier;  
Fonction Premier;  
Début  
    cpt ← 0;  
    i ← 1;  
    TQ (cpt < 10) Faire  
        Dtq  
            Si Premier(i) = vrai Alors  
                cpt ← cpt + 1;  
                Ecrire(i, ' est premier');  
            Sinon  
                i ← i + 1;  
        Fsi  
    Ftq  
Fin;
```



## Exercices 03

✓ Ecrire une procédure « Min » qui retourne le minimum de 3 entiers.

✓ **Solution :**



*Fonction* *Min* (*a,b,c: entier, VAR max: entier*);

*Début*

*max* ← *a*;

*Si b > max Alors max* ← *b*;

*Si c > max Alors max* ← *c*;

*Fin*;





# Exercices 04

- ✓ A partir d'un nombre entier  $N$  on voudrait obtenir deux autres nombres  $N1$  et  $N2$ . Le premier ( $N1$ ) sera constitué par les chiffres pairs de  $N$  et le second ( $N2$ ) par les chiffres impairs.

### Exemples :

➤  $N = 25461327$

$N1 = 2462$

$N2 = 5137$

➤  $N = 42613786$

$N1 = 42686$

$N2 = 137$

➤  $N = 240682$

$N1 = 240682$

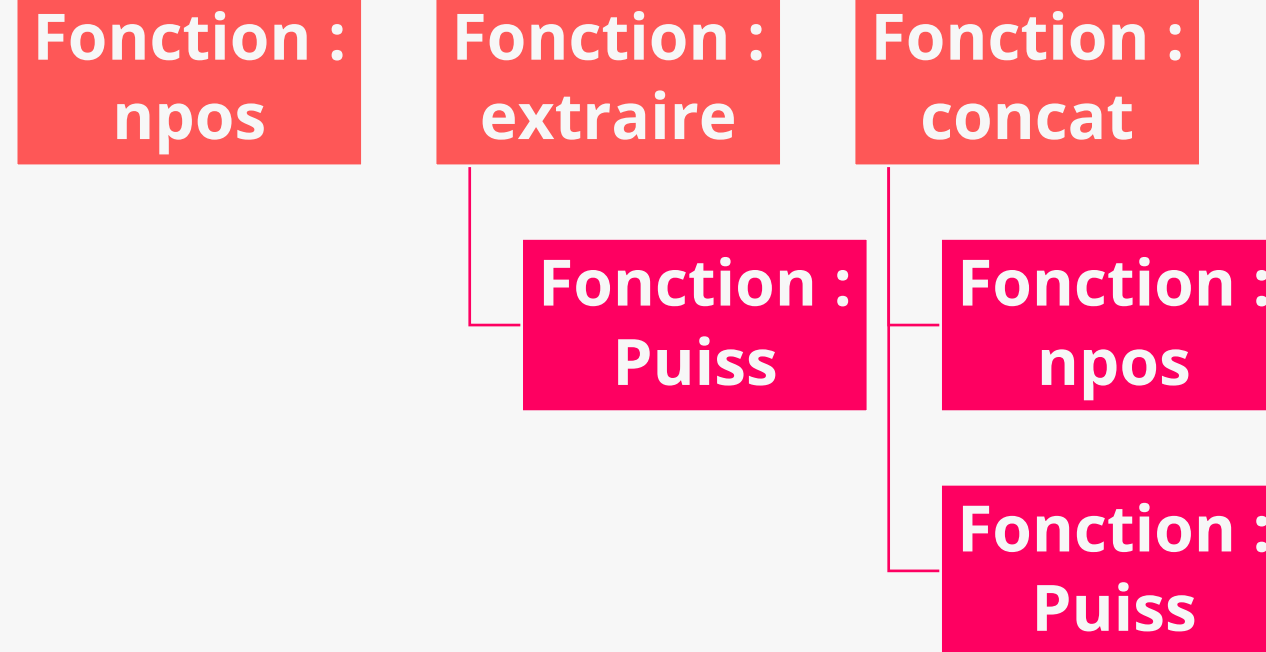
$N2 = 0$

1. Ecrire un algorithme qui permet de retourner les nombre  $N1$  et  $N2$  à partir d'un nombre  $N$ .



# Exercice 04 : Découpage modulaire

Procédure : eclater



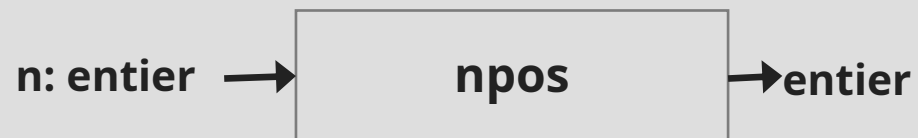
## Exercice 04 : Fonction npos

- ✓ Permet de calculer le nombre de chiffres d'un entier  $n$  (nombre de position).

### Fonction : npos

$\text{npos}(0) = 1$   
 $\text{npos}(2) = 1$   
 $\text{npos}(142) = 3$

### FONCTION



**Rôle :** Retourner le nombre de chiffres de  $n$

```
Fonction npos (n: entier): entier;  
Var P, i: entier;  
Début  
  P ← 0;  
  répter  
    P ← P + 1;  
    n ← n DIV 10;  
  jusqu'à n = 0;  
  npos ← P;  
Fin;
```

## Exercice 04 : Fonction extraire

- ✓ Permet de d'extraire un chiffre (entier) à une position donnée à partir un entier n.

### Fonction : extraire

L'extraction se fait de droite à gauche.

Extraire (1983,3) = 9

Extraire (1983,1) = 3

### FONCTION



**Rôle** : Retourner le chiffre à la position **p** dans **n**

*Fonction* *extraire* (*n,p: entier*): *entier*;

*Fonction* *Puiss*;

*Début*

*extraire* ← (*n DIV Puiss*(10, *p-1*)) *MOD* 10;

*Fin*;

## Exercice 04 : Fonction concat

- ✓ Permet de d'extraire un chiffre (entier) à une position donnée à partir un entier n.

### Fonction : concat

concat (19,83) = 1983  
concat (0,12) = 12  
concat (12,0) = 120

### FONCTION



**Rôle :** concatène les deux entiers **x** et **y**  
(retourne xy)

*Fonction* concat(*x,y: entier*): *entier*;

*Fonction* Puiss, npos;

*Début*

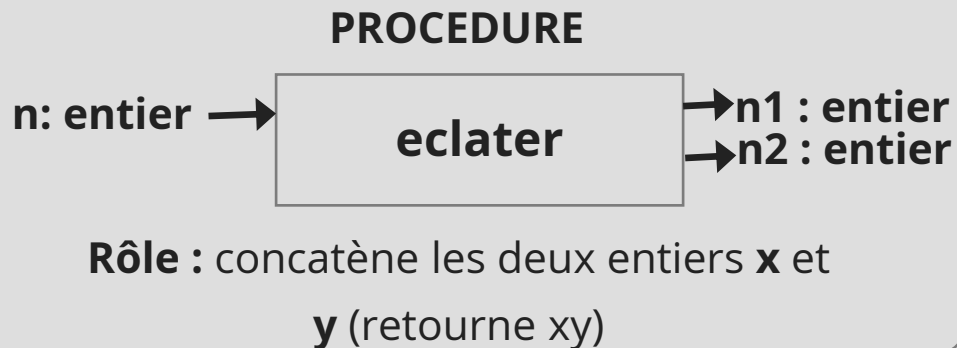
$concat \leftarrow x * Puiss(10, npos(y)) + y;$

*Fin;*

## Exercice 04 : Procédure eclater

- ✓ Permet de d'extraire un chiffre (entier) à une position donnée à partir un entier n.

**Fonction : eclater**



*Procédure eclater(n: entier, VAR n1,n2: entier);*

*Var i, c: entier;*

*Fonction npos, extraire, concat;*

*Début*

*n1 ← 0; n2 ← 0;*

*For i ← 1 à npos(n) Faire*

*Dpour*

*c ← extraire(n,i);*

*Si c MOD 2 = 0 Alors n1 ← concat(c, n1);*

*Sinon n1 ← concat(c, n1);*

*Fsi*

*Fpour*

*n1 ← n1 DIV 10; n2 ← n2 DIV 10;*

*Fin;*