

---

# **TD3 - Les fonctions et les procédures**

Inès de Courchelle - Peio Loubiere



2024-2025

**Objectifs :**

- Manipuler des prédicats
- Connaître la différence entre une fonction et une procédure
- Utiliser des fonctions/procédures dans un programme
- Résoudre des problèmes mathématiques avec des algos

**Consignes :**

- L'ensemble des exercices ci-dessous ne seront pas tous corrigés en cours !
- Les éléments de correction seront donnés en TD, EN AUCUN CAS, des corrections toutes faites vous seront données ou distribuées. Vous devez prendre des notes !
- Durant ce TD, l'utilisation d'un papier et d'un crayon sont fortement conseillés!

**Conseils**

Vous pouvez utiliser d'autres feuilles de notes pour prendre la solution si vous n'avez pas assez de place !

**INTERDIT DE PRENDRE EN PHOTO LES CORRECTIONS AU TABLEAU**

**Durée** 3h

**Format** papier



## Fonction

### Définition

Une fonction est une suite d'instructions qui prend une liste de paramètres en entrée et produit (retourne) un résultat en sortie à l'algorithme appelant cette fonction.

**Attention** Une fonction n'a le droit de retourner qu'une seule et unique information

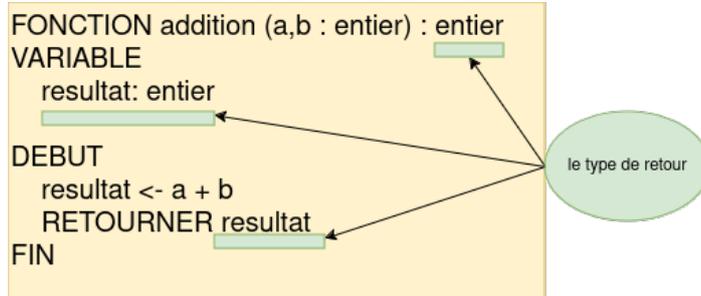
### Représentation



### Exemple



### L'algorithme de la fonction addition



**Attention** Si le type de retour est un entier la variable retournée doit être un entier

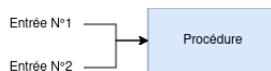


## Procédure

### Définition

Une procédure est un ensemble d'instructions, prenant en entrée des paramètres et qui ne retourne rien. Elle est souvent utilisées pour réaliser des affichages ou réaliser des traitements, sans effets de bords.

### Représentation



### Exemple



### L'algorithme de la procédure bonjour

```
PROCEDURE bonjour (nom,prenom : chaine de caracteres)
DEBUT
  ECRIRE("Bonjour "+prenom +" "+nom+", " +"ça va ?")
FIN
```

PAS DE TYPE DE  
RETOUR

**Attention** Pas de retour !



### OBLIGATOIRE

Pour chaque fonction et procédure, nous définirons :

- **Un résumé** : décrit en une phrase ce que fait la fonction
- **les préconditions** : décrit la nature de l'ensemble des contraintes sur tous les paramètres d'entrée, afin que la fonction puisse s'exécuter correctement
- **la post-condition** : décrit le résultat produit par la fonction/procédures et ses contraintes éventuelles

**Où ?** En haut du programme

#### addition

```
//Fonction qui additionne 2 entiers en paramètres
// Préconditions : 2 entiers quelconque
// Postconditions : retourne l'addition des 2 entiers
FONCTION addition (a,b : entier) : entier
VARIABLE
  resultat : entier
DEBUT
  resultat <- a + b
RETOURNER resultat
FIN
```

#### bonjour

```
// Procédure qui affiche un bonjour à une personne
// Préconditions : 2 chaînes de caractères (nom et prénom)
// Postconditions : affiche "bonjour" avec le nom et le prénom
passés en paramètre
PROCEDURE bonjour (nom,prenom : chaîne de caracteres)
DEBUT
  ECRIRE("Bonjour "+prenom +" "+nom+", " + "ça va ?")
FIN
```

## Exo 1 : Il fait trop chaud pour travailler



Écrire une fonction qui permet de convertir une température, exprimée en °C, en °F.



**Pour rappel :**  $T(F) = 9 \div 5 \times T(C) + 32$



## Exo 2 : PGCD



1. Faut-il utiliser une fonction ou une procédure ? justifiez votre réponse
2. Écrire l'algorithme de la fonction ou procédure qui calcule le PGCD avec l'algorithme d'Euclide
3. Écrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche leur PGCD



### Rappel du PGCD

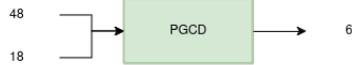
Le **Plus Grand Commun Diviseur** est le plus grand nombre qui divise à la fois a et b.

### Exemple

$$\text{PGCD}(48,18)= 6 \quad \text{PGCD}(14,17)= 1$$

### Algorithme d'Euclide

On utilise la méthode des divisions successives



Pour trouver le PGCD de 48 et 18, nous devons réaliser les étapes suivantes :

1.  $48 \div 18 = 2$  et il reste 12
2.  $18 \div 12 = 1$  et il reste 6
3.  $12 \div 6 = 2$  et il reste 0

Le dernier diviseur est le PGCD lorsqu'il reste 0.

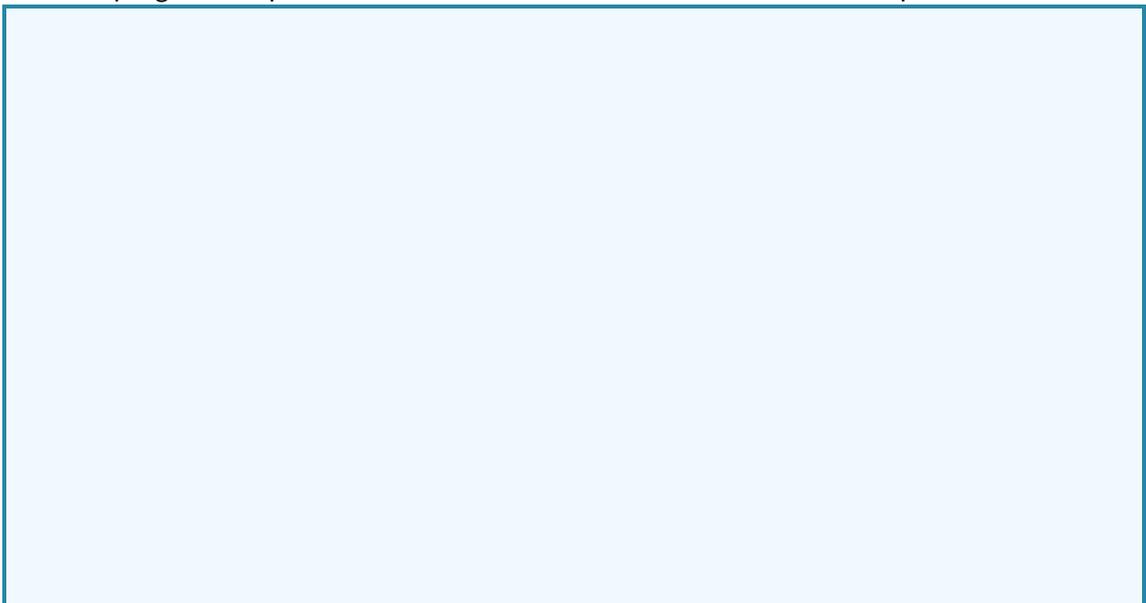
$$\text{PGCD}(48,18)= 6$$

1. Faut-il utiliser une fonction ou une procédure ? justifiez votre réponse

2. Écrire l'algorithme de la fonction ou procédure qui calcule le PGCD avec l'algorithme d'Euclide



3. Écrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche leur PGCD



### Exo 3 : Nb premier



1. Qu'est ce qu'un prédicat ?
2. Écrire un prédicat qui vérifie qu'un nombre est premier
3. Écrire un programme qui permet d'afficher tous les nombres premiers inférieurs à un entier donné.

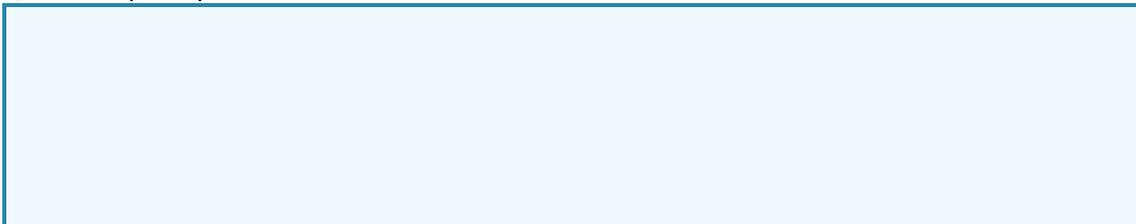


**Définition** Un nombre entier est un nombre premier s'il n'est divisible que par 1 et par lui-même.

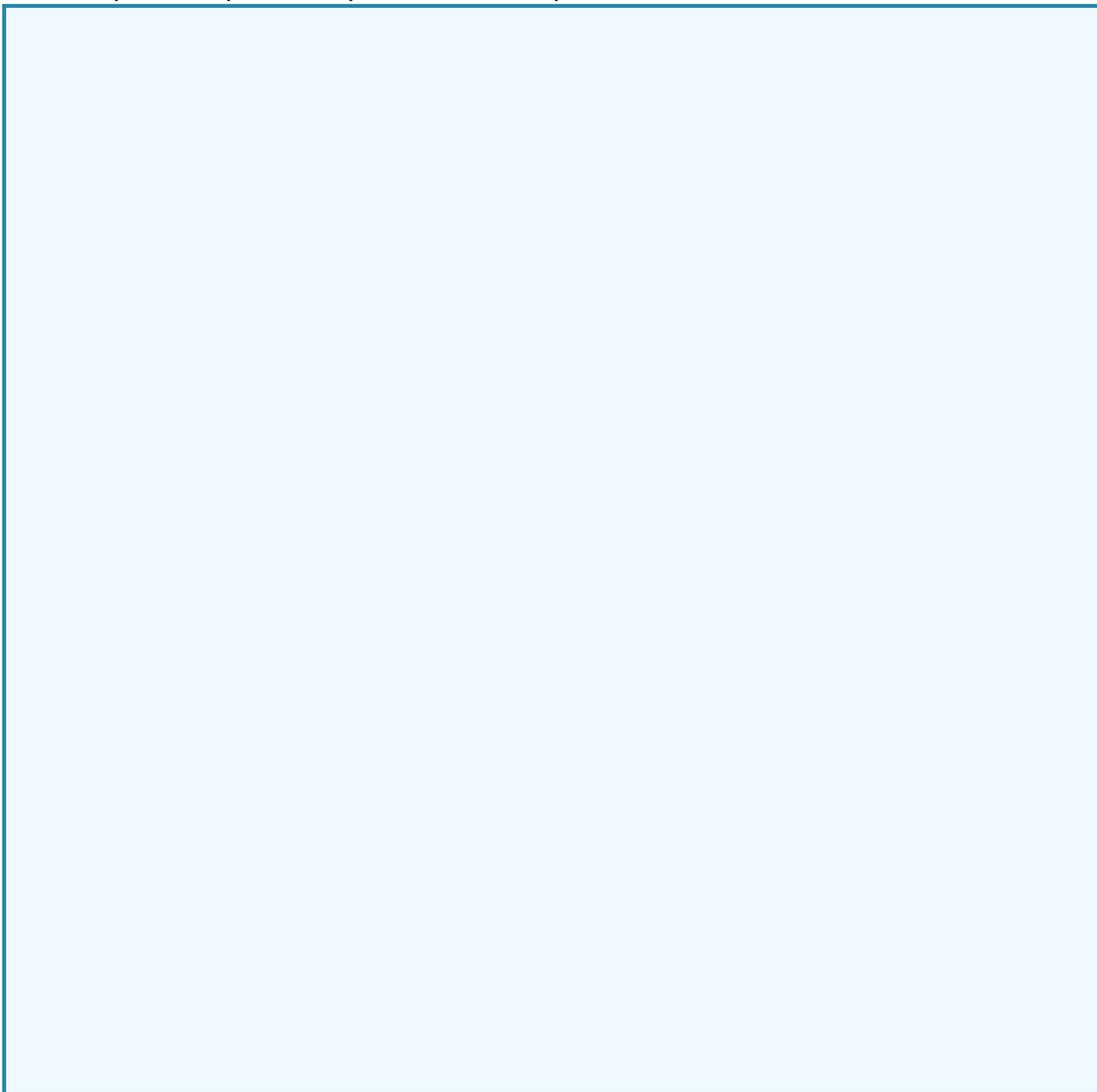
**La liste**

0									
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

1. Qu'est ce qu'un prédicat ?



2. Écrire un prédicat qui vérifie qu'un nombre est premier.



3. Écrire une procédure qui permet d'afficher tous les nombres premiers inférieurs à un entier donné.



#### Exo 4 : Maxis

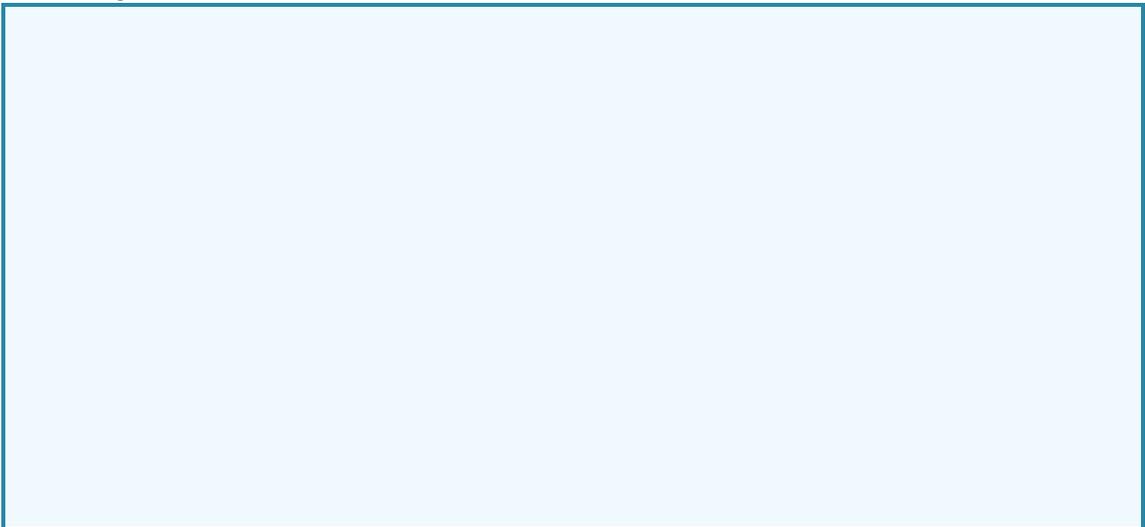


1. Ecrire l'algorithme d'une fonction maxi qui calcule le maximum entre 2 nombres
2. Ecrire l'algorithme d'une fonction maxi3 qui calcule le maximum entre 3 nombres

1. Ecrire l'algorithme d'une fonction maxi qui calcule le maximum entre 2 nombres



2. Ecrire l'algorithme d'une fonction maxi3 qui calcule le maximum entre 3 nombres



## Exo 5 : Exposant



1. Écrire l'algorithme d'une fonction qui prend en paramètre un nombre réel et un exposant entier et met le nombre à l'exposant défini
2. Écrire l'algorithme du programme permettant de tester votre fonction

1. Écrire l'algorithme d'une fonction qui prend en paramètre un nombre réel et un exposant entier et met le nombre à l'exposant défini

2. Écrire l'algorithme du programme permettant de tester votre fonction

## Exo 6 : On s'entraîne sur les chaînes de caractères...



1. Écrire une fonction `prCentVoy` qui compte le pourcentage de voyelles d'une chaîne de caractères.
2. Écrire une fonction `miroir` qui inverse une chaîne de caractères **miroir**("toto")  
**retourne "otot"**
3. Écrire un prédicat `estPalindrome` qui vérifie qu'une chaîne de caractères est un palindrome (sans espace).
4. Écrire un prédicat `estPalindromeEspace` qui vérifie qu'une chaîne de caractères est un palindrome (avec espace).



### Palindrome

Pour rappel, un palindrome est une suite de caractères qui peut se lire dans les deux sens : "non", "kayak", "Esope reste ici et se repose" sont des palindromes.

### Fonctions utiles

#### Fonctions numériques

- `ent` : retourne la partie entière d'un réel.
- `aléa` : retourne un nombre aléatoire entre 0 et n exclu
- `sqrt` : retourne la racine carré d'un réel positif

#### Fonction textuelles

- `longueur` : retourne le nb de caractères d'une chaîne
- `extrait` : retourne une sous-chaîne comprise entre deux indices (on commence à 0)

**Exemple**

```
1 PROGRAMME EXEMPLE
2 VARIABLES
3   a,b,taille : entier
4   concat,prenom,uneChaine, abrege : chaine de caractères
5
6 DEBUT
7   a <- 4,45
8   b <- ent(a) // b sera égal à 4
9   nbAlea <- alea(0,10) // nbAlea sera égal à un nombre alé
   atoire entre 0 et 9
10  maRacine <- sqrt(16) // maRacine sera égal à 4
11
12  prenom <- "Maeva"
13
14  taille <- longueur(prenom) // taille sera égal à 5
15
16  uneChaine <- "Bonjour tu vas bien ?"
17
18  abrege <- extrait(0,4,uneChaine)
19  // abrege sera égal à "Bonjo"
20
21  concat <- uneChaine + " très bien et toi ?"
22  // concat sera égale à "Bonjour tu vas bien ? très bien
   et toi ?"
23
24
25 FIN
```

1. Écrire une fonction `prCentVoy` qui compte le pourcentage de voyelles d'une chaîne de caractères.

2. Écrire une fonction miroir qui inverse une chaîne de caractères :



3. Écrire un prédicat estPalindrome qui vérifie qu'une chaîne de caractères est un palindrome (sans espace).



4. Écrire un prédicat `estPalindromeEspace` qui vérifie qu'une chaîne de caractères est un palindrome (avec espace).

## Exo 7 : La somme des chiffres

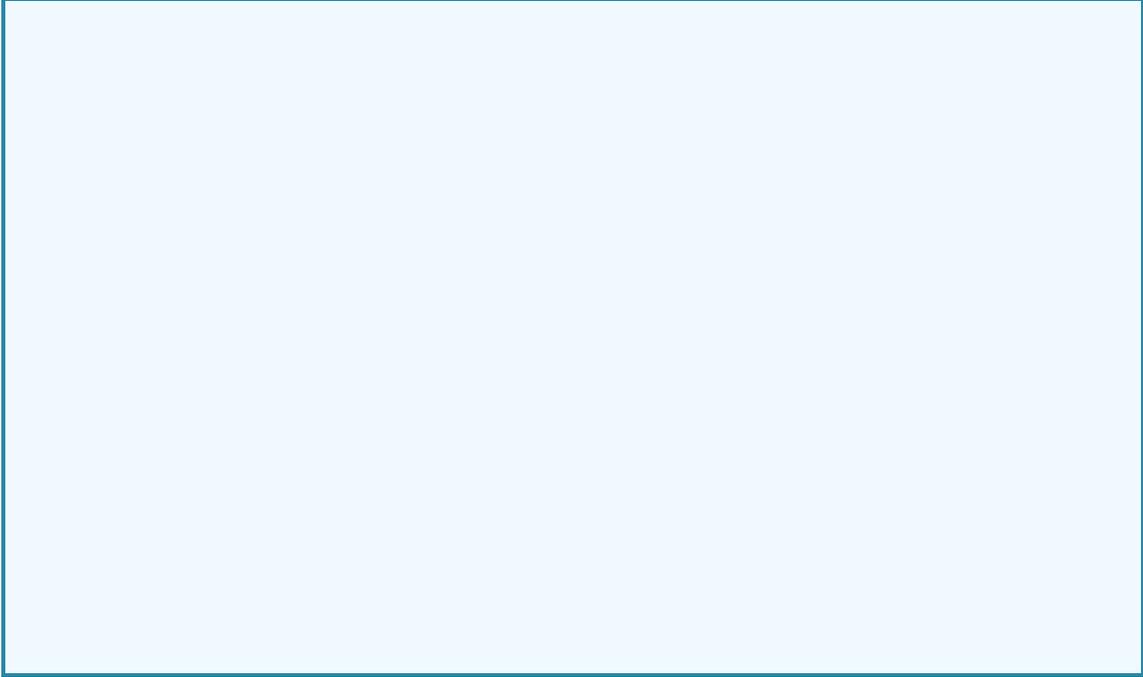


1. Écrire une fonction `somChiffres` qui calcule la somme des chiffres qui composent un nombre positif donné, le résultat est un chiffre.

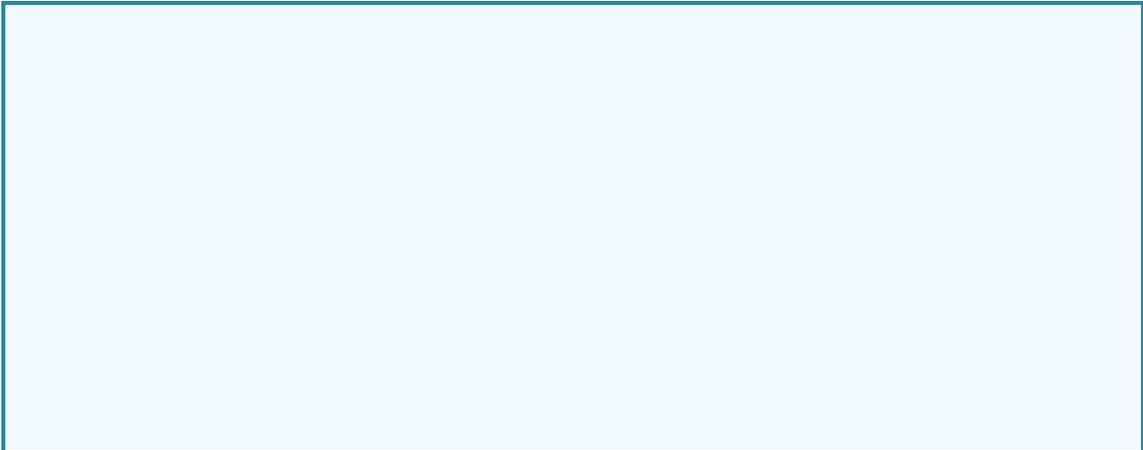
`somChiffres(3) = 3`, `nbChiffres(2021) = 5`, `nbChiffres(666) = 9`

2. Écrire le programme principal qui permet d'utiliser votre fonction

1. Écrire une fonction `somChiffres` qui calcule la somme des chiffres qui composent un nombre positif donné, le résultat est un chiffre.



2. Écrire le programme principal qui permet d'utiliser votre fonction



## Exo 8 : Inclusion



Écrire un prédicat `estIncluse` qui vérifie qu'une chaîne de caractères est bien contenue dans une seconde.



Ce problème a été adressé efficacement par différents algorithmes, notamment celui de Knuth-Morris-Pratt ou encore celui de Boyer-Moore-Horspool