

---

## **TD2 - Les éléments de base**

Inès de Courchelle - Peio Loubiere



2024-2025

**Objectifs :**

- Comprendre la structure d'un algorithme
- Connaître la différence entre une boucle POUR et une boucle TANT QUE
- Utiliser la notation conditionnelle dans un algorithme
- Manipuler les entrées/sorties dans un algorithme (LIRE, ÉCRIRE)

**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** 3h00

**Format** papier

**Exo 1 : ¿ Qué pasa ?**

```
1 PROGRAMME test1
2 VARIABLES
3   a,b : entier
4   c : réel
5 DEBUT
6   a ← 10
7   b ← 2
8   c ← (a+3)/2
9   ECRIRE("a=" + a + "b=" + b + "c=" + c)
10  b ← b DIV 2
11  a ← a MOD 4
12  c ← b/a
13  ECRIRE("a=" + a + "b=" + c + "c=" + b)
14  ECRIRE("Fin du programme")
15 FIN
```



Que fait ce programme ?

## Exo 2 : Easy Peasy



1. Écrire un algorithme qui demande la saisie d'un nombre et affiche son carré
2. Écrire un algorithme qui demande la saisie de 2 nombres entiers et affiche leur moyenne
3. Ecrire un algorithme qui calcule et affiche l'aire d'un rectangle dont les dimensions (longueur et largeur) sont saisies par l'utilisateur



### Programme :

un programme est l'unité de base d'un algorithme, il possède un DEBUT et une FIN et a une syntaxe de base :

```
1 PROGRAMME NomDuProgramme
2 DEBUT
3 // commentaire sur une ligne
4 /*
5 Commentaire sur
6 plusieurs lignes
7 */
8 FIN
```

**Variable :**

- élément du programme qui sert à stocker une valeur, un résultat de calcul
- définie par un nom et un type
- une variable ne peut avoir qu'un seul type
- d'abord déclarée dans la zone de déclaration,
- puis initialisée dans la zone d'instructions
- et utilisée dans la zone d'instructions

**ECRIRE :**

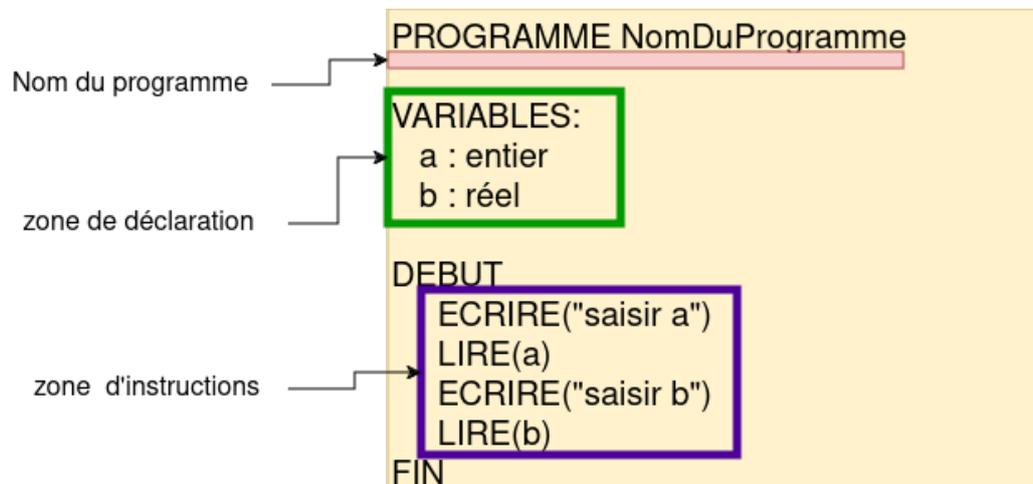
Permet d'afficher une instruction ou un message à l'écran

```
1 ECRIRE("Un message")
```

**LIRE :**

Permet de récupérer une saisie clavier de l'utilisateur

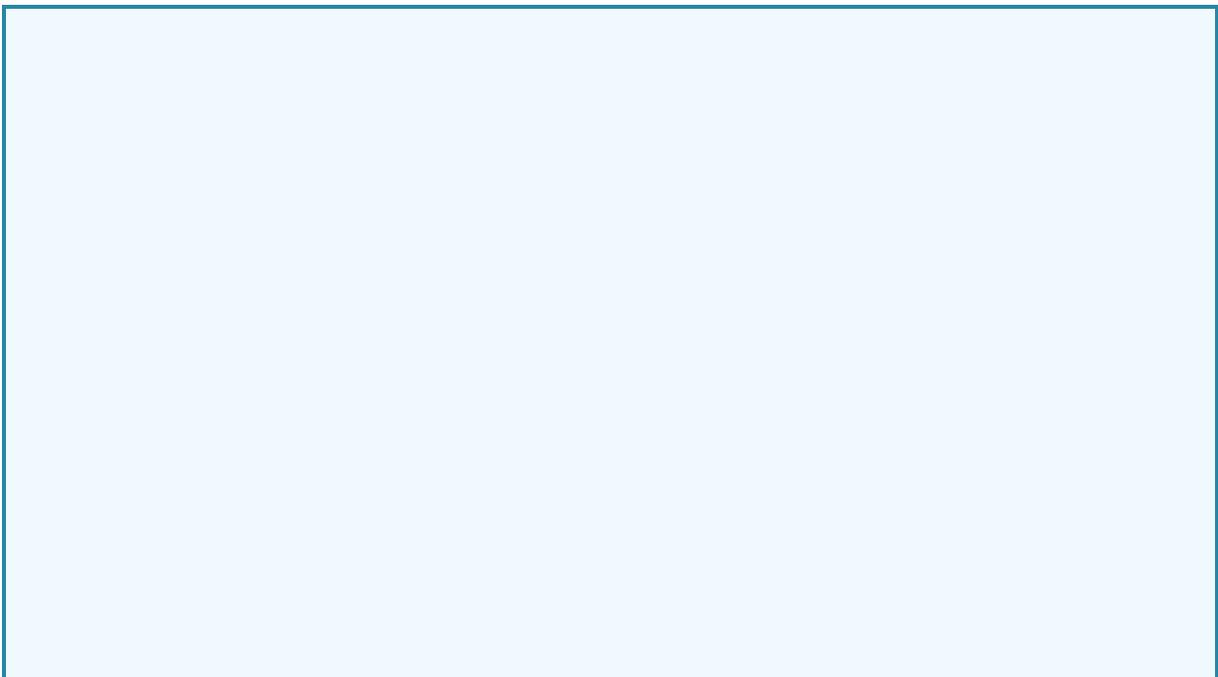
```
1 LIRE(X)
```

**Exemple :**

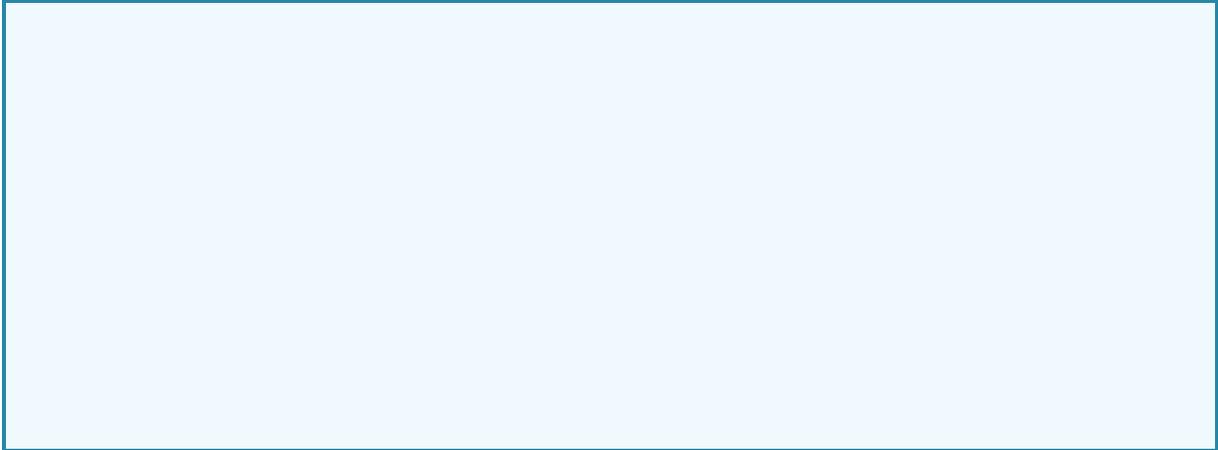
1. Écrire un algorithme qui demande la saisie d'un nombre et affiche son carré



2. Écrire un algorithme qui demande la saisie de 2 nombres entiers et affiche leur moyenne



3. Ecrire un algorithme qui calcule et affiche l'aire d'un rectangle dont les dimensions (longueur et largeur) sont saisies par l'utilisateur

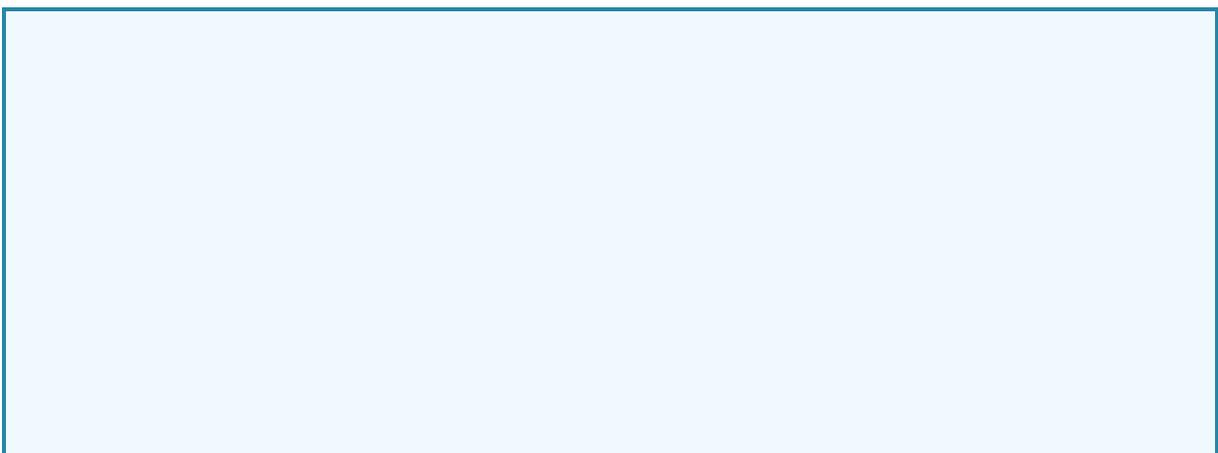


### Exo 3 : La TVA



L'impôt sur la TVA est de 20% sur les produits. Écrire un algorithme qui :

- Demande à l'utilisateur un prix de base et calcule son prix total en ajoutant la TVA
- Affiche un message faisant apparaître le prix de base, la TVA, et le prix final



## Exo 4 : Le max du max



Écrire un algorithme qui affiche le maximum de 3 nombres saisis par l'utilisateur.  
*Il existe plusieurs méthodes, qui fonctionnent, mais dans tous les cas, essayez de ne pas faire 3 SI différents !*



### Instructions conditionnelles

#### Le si

```
1 SI (expression booléenne) ALORS
2   suite d'instructions
3 FIN SI
```

#### Le si ... sinon ...

```
1 SI (expression booléenne) ALORS
2   suite d'instructions
3 SINON
4   suite d'instructions
5 FIN SI
```

#### Attention

Pour un si ... sinon il n'y a qu'un seul FIN SI

#### Expression booléenne

##### Définition

Une expression qui donne une valeur booléenne (vrai ou faux)

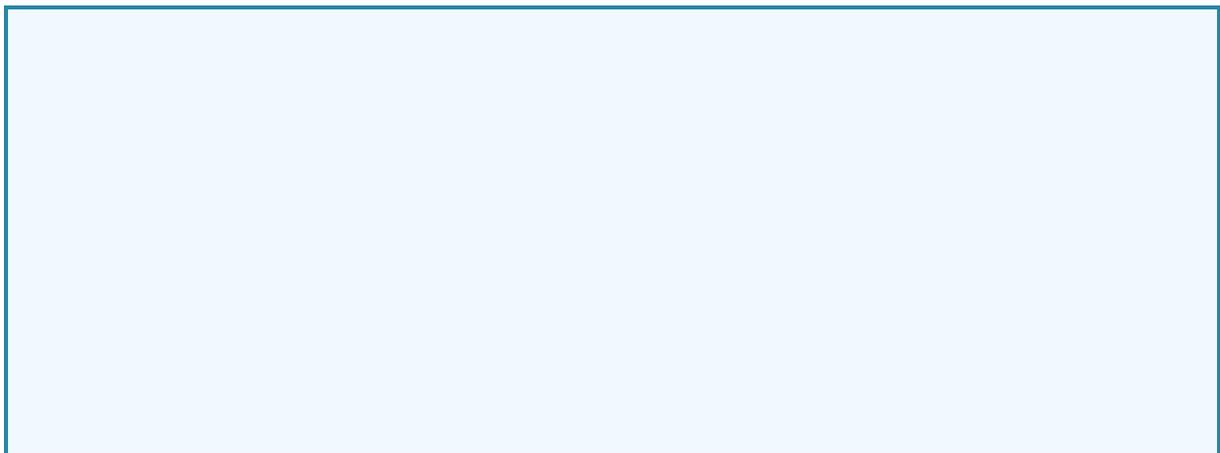
##### Exemple

La valeur pour  $42 > 24$  est VRAI. 42 est bien supérieur à 24 !

1. Méthode 1 : Le "si" "sinon si"



2. Méthode 2 : L'utilisation de deux "si" avec une variable temporaire



## Exo 5 : Une note



Écrire un algorithme qui :

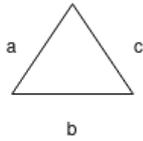
- Demande à l'utilisateur de saisir une note (un nombre compris entre 0 et 20)
- Affiche un message en fonction de la note :
  - "Ajourné" si la valeur est inférieure ou égale à 6.
  - "Rattrapage" si la valeur est supérieure à 6 et inférieure à 10.
  - "Admis" si la valeur est supérieure ou égale à 10.
  - "ERREUR" si la note est totalement en dehors de l'intervalle [0,20].

## Exo 6 : Le triangle



Écrire un algorithme qui :

- Demande à l'utilisateur de saisir les longueurs des côtés d'un triangle en commençant par la plus grande.
- Détermine si ce triangle est bien un triangle et s'il est équilatéral, isocèle, rectangle ou quelconque.



Un triangle est rectangle si  $a = \sqrt{(b \times b) + (c \times c)}$



## Exo 7 : La calculatrice



Écrire un algorithme qui :

- Demande à l'utilisateur de saisir deux nombres réels a et b qui seront les opérandes de l'opération à effectuer.
- Demande à l'utilisateur de saisir l'opérateur souhaité (+, -, / ou \*).
- Affiche le résultat de l'opération ou bien un message d'erreur si la donnée saisie ne correspond pas à un opérateur.

**Attention : La donnée saisie est ici un caractère, pas un opérateur, on ne peut pas l'utiliser telle quelle !**



## Exo 8 : Je tourne en rond !

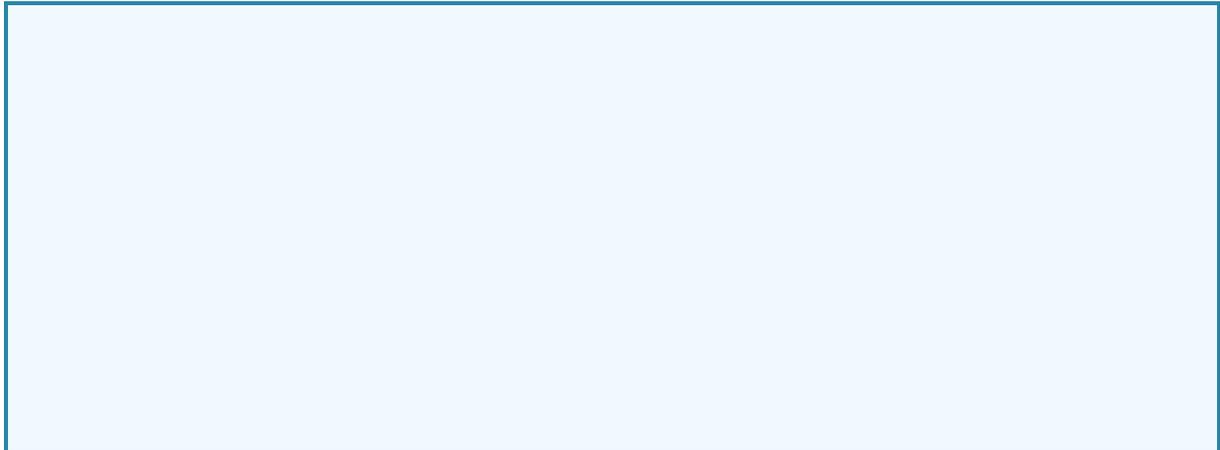


On veut déterminer si un point d'un plan est situé dans un cercle. On considère un cercle donc les coordonnées du centre sont  $(0,0)$  et son rayon  $r$ . 1. Écrire un algorithme qui :

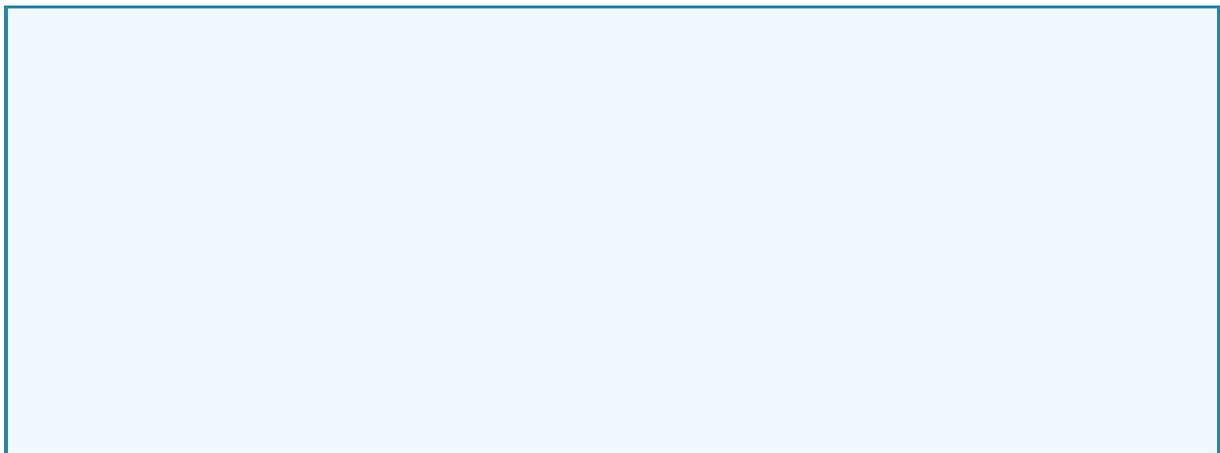
- Demande à l'utilisateur de choisir un réel  $r$ .
  - Demande à l'utilisateur de rentrer les coordonnées d'un point  $x$  et  $y$ .
  - Affiche si ce point appartient au cercle, et en dehors du cercle ou est sur le cercle.
2. Modifier l'algorithme précédent pour que l'utilisateur puisse également choisir les coordonnées du centre du cercle

1. Écrire un algorithme qui :

- Demande à l'utilisateur de choisir un réel  $r$ .
- Demande à l'utilisateur de rentrer les coordonnées d'un point  $x$  et  $y$
- Affiche si ce point appartient au cercle, et en dehors du cercle ou est sur le cercle.



2. Modifier l'algorithme précédent pour que l'utilisateur puisse également choisir les coordonnées du centre du cercle



## Exo 10 : Les tables de multiplication



Écrire un algorithme qui :

- Demande à l'utilisateur de saisir un nombre entier  $n$
- Affiche la table de multiplication de  $n$  (ses multiples entre 0 et 10)



### La boucle **POUR**

**Quand ?** Les boucles **POUR** sont utilisés quand on connaît le nombre de tours par avance

**Comment ?**

```
1 POUR i allant de 1 à 10 FAIRE
2     ECRIRE("Bonjour")
3 FIN POUR
```

### Mise en garde

- On ne modifie jamais les bornes de début et de fin dans la boucle
- On ne modifie jamais la valeur de l'itérateur dans la boucle
- On n'arrête jamais un pour avant sa fin prévue

*si vous avez besoin de transgresser un de ces principes, c'est que vous ne devez pas utiliser une boucle pour*



### La boucle TANT QUE et REPETER ... TANT QUE

**Quand ?** Les boucles **TANT QUE** sont utilisées quand on ne sait pas par avance combien de fois on va répéter la série d'instructions

**Comment ?**

#### Exemple 1

```
1 choix <- VRAI
2 TANT QUE (choix) FAIRE
3     ECRIRE("Bonjour")
4 FIN TANT QUE
```

#### Exemple 2

```
1 choix <- VRAI
2 REPETER
3     ECRIRE("Bonjour")
4 TANT QUE (choix)
```

#### Différence entre TANT QUE et REPETER



## Exo 11 : Somme des premiers entiers



Écrire un algorithme qui :

- Demande à l'utilisateur de saisir un nombre entier positif  $n$
- Vérifie si le nombre entré est bien positif
- Affiche si un nombre entier  $a$ , saisi par l'utilisateur, est premier ou non.
- Affiche la somme de tous les nombres entiers entre  $a$  et  $n$



**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



## Exo 12 : Le Compte à rebours



Écrire un algorithme qui :

- Demande à l'utilisateur de saisir un nombre de secondes
- Convertit ce nombre au format en hh : min : s
- Décrémente les secondes (affiche à chaque changement, l'heure sous format hh : min : s)
- S'arrête lorsque l'on atteint 00 : 00 : 00 et affiche "BOOM"

**DIV**

L'opérateur DIV permet de réaliser une division

**Exemple**

```
1 m <- 180 DIV 60
2 /* m sera égal à 3 */
```

**MOD**

L'opérateur MOD permet d'obtenir le reste de la division entière.

**Exemples**

- $97 \bmod 60 = 37$  Explications  $97 \div 60 = 1$  et il reste **37**
- $545 \bmod 60 = 9$  Explications  $545 \div 60 = 9$  et il reste **5**

## Exo 13 : Factorielle



Écrire un algorithme qui :

- Demande à l'utilisateur de saisir un entier
- Calcule la factorielle de cet entier
- Affiche la factorielle



On rappelle que la factorielle d'un entier naturel  $n$  (notée  $n!$ ). Elle se calcule de la manière suivante:  $n! = 1 \times 2 \times 3 \times 4 \times 5 \times \dots \times n$  et  $0! = 1$ .

## Exo 14 : 12345



Écrire un algorithme qui affiche le plus petit entier  $n$  tel que  $(n+1)^*(n+3)$  dépasse 12345.



### Exo 15 : Le plus grand nombre



Écrire un algorithme qui :

- Demande à l'utilisateur d'entrer une série d'entiers
- Affiche le plus grand d'entre eux, ainsi que la position à laquelle il a été entré

La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro ou un entier impair multiple de cinq.

**Attention : vous n'avez pas besoin de tableau pour réaliser cette exercice**

#### Exemple de fonctionnement

```
1 Entrez le nombre numéro 1: 17
2 Entrez le nombre numéro 2: 6
3 Entrez le nombre numéro 3: -1
4 Entrez le nombre numéro 4: 22
5 Entrez le nombre numéro 5: 15
6 Le plus grand nombre est 22, saisi en numéro 4
```

