TD 1 - Rappels

Inès de Courchelle, Elisabeth Ranisavljevic, Romain Dujol

2024-2025

Objectifs :

- Faire un point sur toutes les notions vues l'année dernière
- Réaliser des rappels autour de la récursivité
- Partir sur de bonnes bases

Durée 1h30
Format papier
Attention Tous les exercices ne seront pas corrigés en cours. On vous en laisse pour vos révisions !

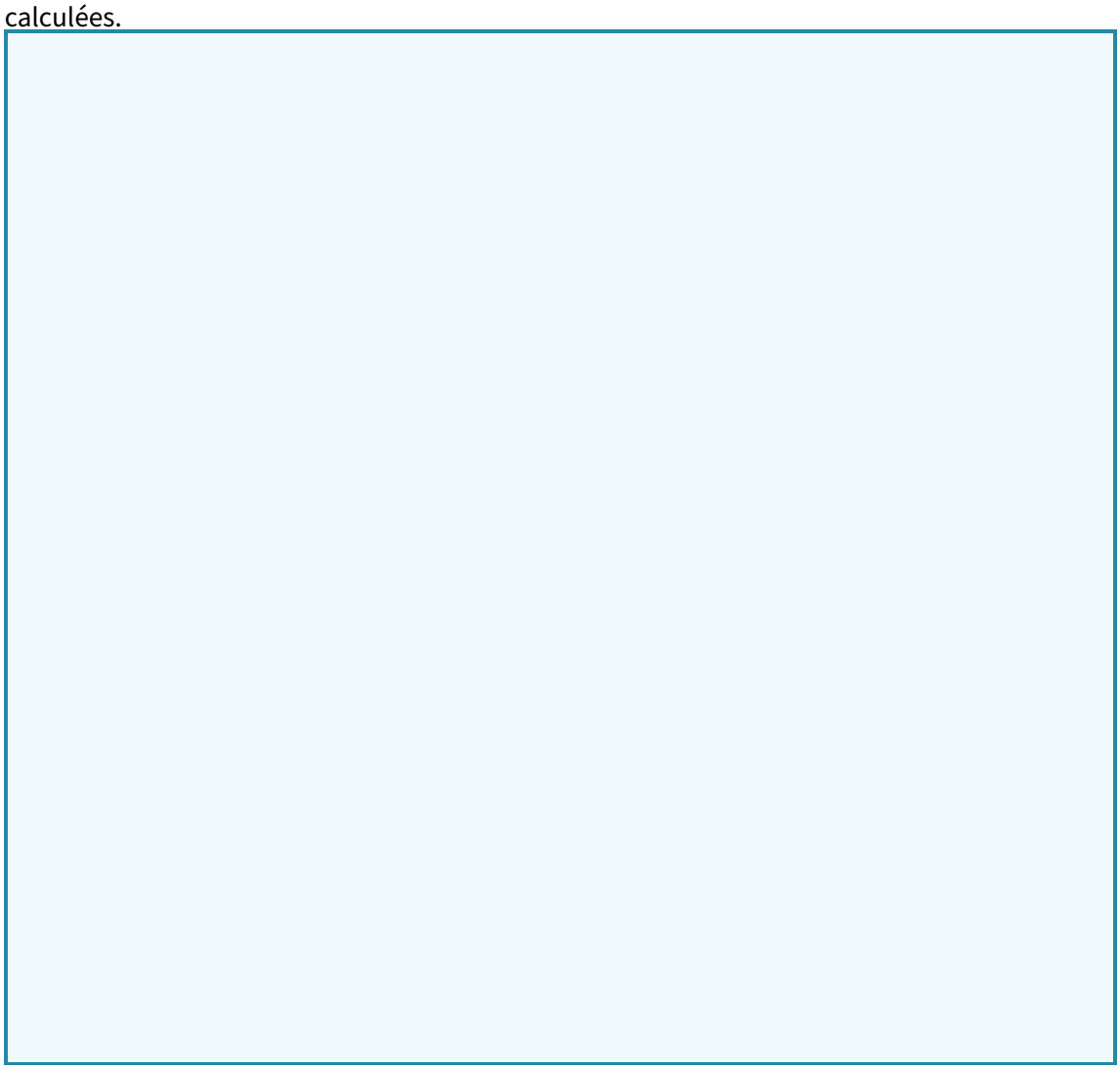
Algo de base - sans récursivité !

Exercice 1 : Le diviseur

1. Écrire une procédure divmod qui calcule le quotient et le reste de la division de deux entiers positifs (sans utiliser la récursivité). La procédure affichera le quotient et le reste.
2. Écrire l'algorithme du programme demandant la saisie de 2 entiers. Il affichera les 2 valeurs calculées.

1. Écrire une procédure divmod qui calcule le quotient et le reste de la division de deux entiers positifs (sans utiliser la récursivité). La procédure affichera le quotient et le reste.

2. Écrire l'algorithme du programme demandant la saisie de 2 entiers. Il affichera les 2 valeurs calculées.



Exercice 2 : le PGCD

1. Écrire l'algorithme de la fonction ou procédure qui calcule le PGCD.
2. Écrire un programme qui demande à l'utilisateur de saisir deux nombres et qui affiche leur PGCD

Rappel du PGCD

Le **P**lus **G**rand **C**ommun **D**iviseur est le plus grand nombre qui divise à la fois a et b.

Exemple

PGCD(48,18)= 6 PGCD(14,17)= 1

Algorithme d'Euclide

On utilise la méthode des divisions successives

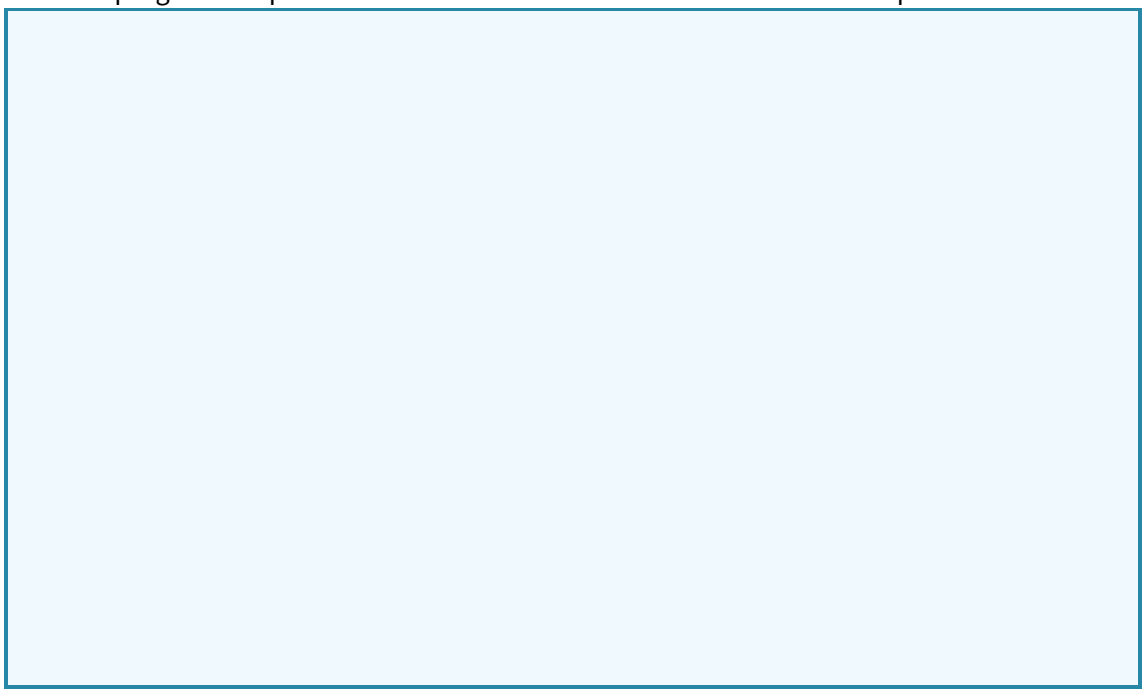
Pour trouver le PGCD de 48 et 18, nous devons réaliser les étapes suivantes :

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.
PGCD(48,18)= 6

1. Écrire l'algorithme de la fonction ou procédure qui calcule le PGCD.

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



Exercice 3 : nb premier

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.

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.

La récursivité

Exercice 1 : La puissance

1. Écrire une fonction récursive non terminale, qui calcul a à la puissance b, comme suit :

```
Fonction puissanceRec ( a , b : entier ) : entier
```

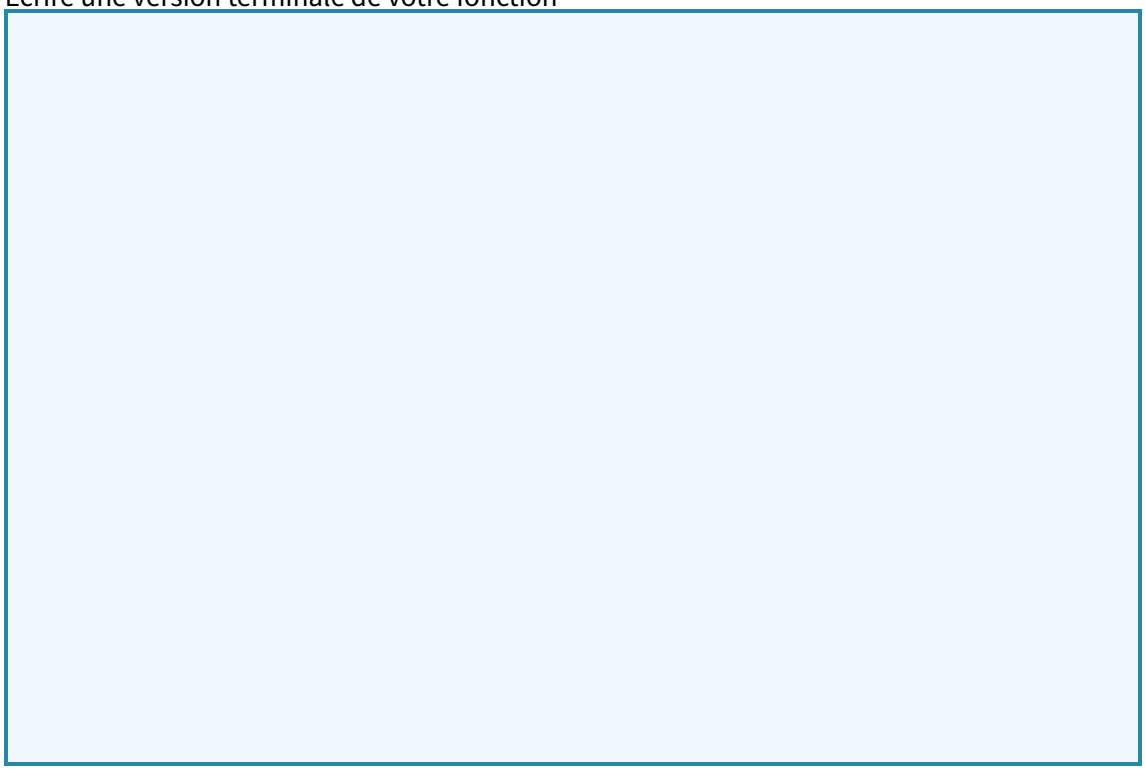
2. Dessiner l'arbre de récursivité de puissanceRec(3,4)
3. Écrire une version terminale de votre fonction
4. Dessiner l'arbre de récursivité de votre fonction terminale avec 3 à la puissance 4.

1. Écrire une fonction récursive non terminale, qui calcul a à la puissance b, comme suit :

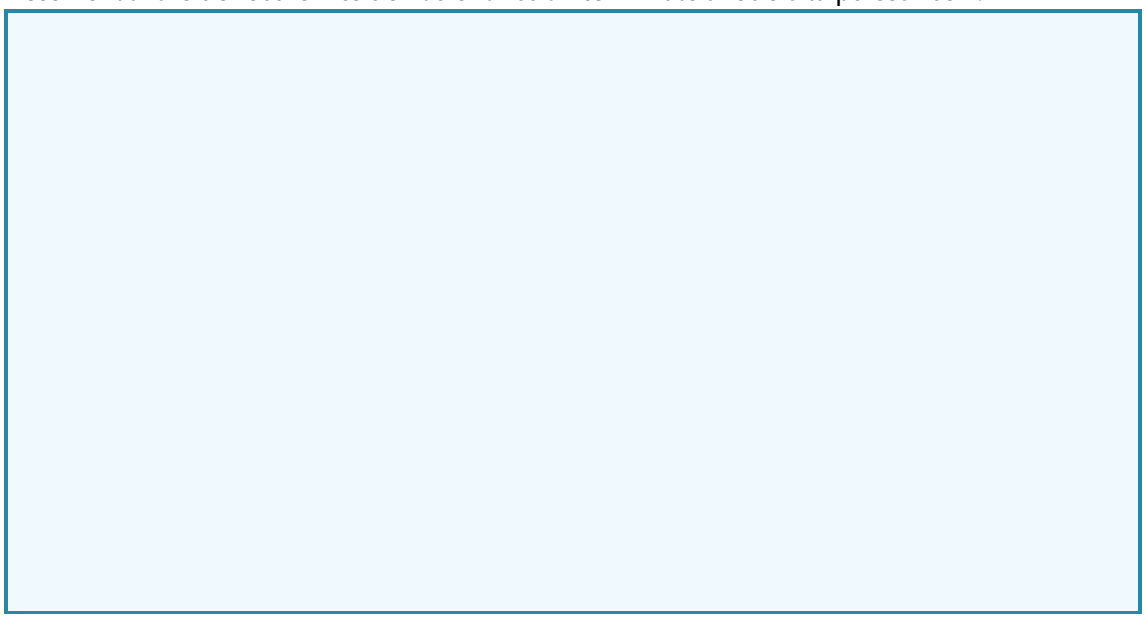
```
Fonction puissanceRec ( a , b : entier ) : entier
```

2. Dessiner l'arbre de récursivité de puissanceRec(3,4)

3. Écrire une version terminale de votre fonction



4. Dessiner l'arbre de récursivité de votre fonction terminale avec 3 à la puissance 4.



Exercice 2 : Somme des chiffres

Écrire une fonction récursive qui retourne la somme des chiffres d'un nombre composé de n chiffres.

```
1 Fonction SommeChiffre ( a : entier ) : entier
```

Exemple : sommeChiffre(2195) = 2+1+9+5 = 17

Exercice 3 : Palindrome

Un palindrome est une chaîne de caractères qui se lit de gauche à droite ou de droite à gauche, comme “kayak”, “radar”, “elle”, “été”, “esope reste ici et se repose”, “engage le jeu que je le gagne” (Pierre Bailly), “l’ame soeur, elle, rue, ose mal” (Alain Damasio). Écrire un prédicat récursif qui indique si un mot simple est un palindrome.

Rappel

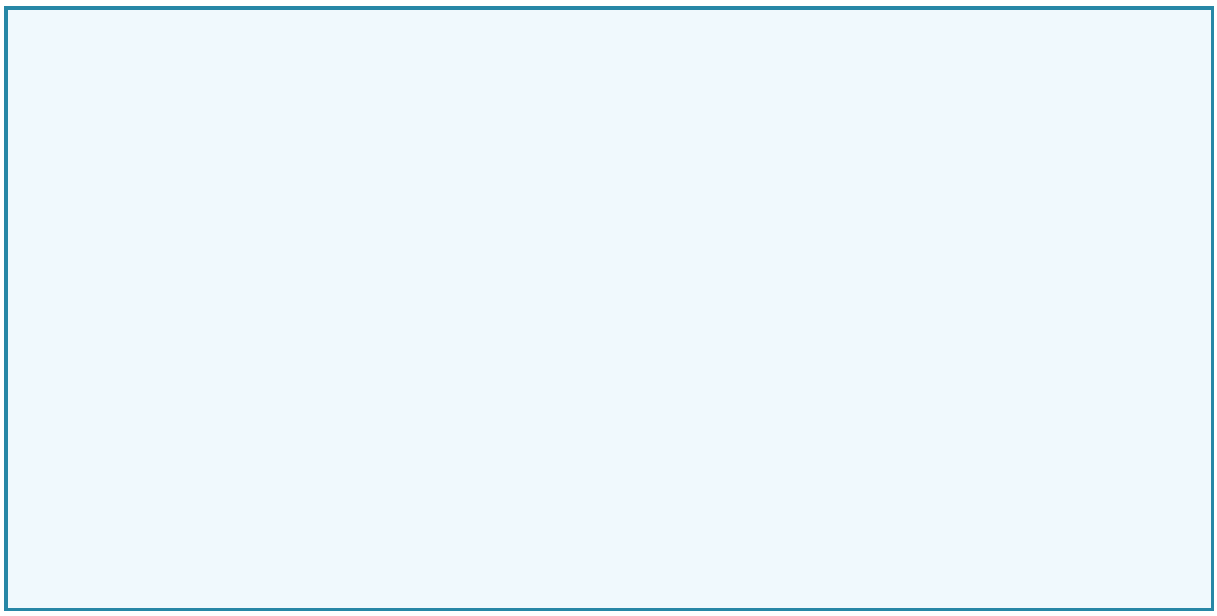
On peut utiliser la fonction
`extrait(ch:chaine de car,i :entier ,j: entier): chaine de car`

Exemple 1

```
ch1 <- "Jordan"
ch2 <- extrait(ch1,1,1)
/* resultat */
/* ch2 = 'o' */
```

Exemple 2

```
ch1 <- "Jordan"
ch2 <- extrait(ch1,0,3)
/* resultat */
/* ch2 = 'Jord' */
```



La complexité

Exercice 1

Donner l'ordre de complexité des codes C suivants (en fonction des variables n et m). Attention, justifiez vos réponses.

1. Code 1 :

```
PROGRAMME CODE1
VARIABLES
    i,x,j : entier
DEBUT
  x <- 1
  POUR i allant de 1 à N FAIRE
     x <- x + 1
  FIN POUR

  POUR j allant de 1 à N FAIRE
    x <- x + 1
  FIN POUR
FIN
```

2. Code 2 :

```
PROGRAMME CODE2
VARIABLES
    i,x,j : entier
DEBUT
  x <- 1
  POUR i allant de N à N+6 FAIRE
    POUR j allant de M à M+3 FAIRE
        x <- x + 1
    FIN POUR
  FIN POUR
FIN
```

3. Code 3 :

```
PROGRAMME CODE3
VARIABLES
    i,x,j : entier
DEBUT
    x <- 1
    i <- 1
    j <- 1

    TANT QUE (i <= N ET j <= M) FAIRE
        x <- x + 1
        i <- i + 1
        j <- j + 1
    FIN TANT QUE

    TANT QUE (i <= N) FAIRE
```

```
16          x <- x + 1
17          i <- i + 1
18      FIN TANT QUE
19
20      TANT QUE (j <= M) FAIRE
21          x <- x + 1
22          j <- j + 1
23      FIN TANT QUE
24  FIN
```

Exercice 2

Nous considérons le code suivant :

```
1   PROGRAMME EXO2
2   VARIABLES
3       i,s : entier
4       T : tableau d'entiers de taille N
5   DEBUT
6       /* T est un tableau d'entiers de taille N dont les valeurs
7       de T sont initialisées aléatoirement entre 1 et 100 inclus  */
8       T <- initialiser(N)
9
10      s <- 0
11
12      POUR i allant de 0 à N FAIRE
13          SI (T[i]*T[i] > 100) ALORS
14              s <- s + T[i]
15          FIN SI
16      FIN POUR
17  FIN
```

Quel est le nombre **d'additions** réalisées par cet algorithme dans le cas moyen, en supposant les valeurs de T sont aléatoires entre 1 et 100 inclus.

Exercice 3

Nous considérons le code suivant :

```
PROGRAMME EXO3
VARIABLES
    i,s : entier
    T : tableau de booleens de taille N
DEBUT
   /* T est un tableau de booleen de taille N dont les valeurs
    de T sont initialisées aléatoirement avec VRAI ou FAUX  */
    T <- initialiser(N)

    s <- 0
    i <- 0

    TANT QUE (i < N-1) FAIRE
        SI (T[i] ET T[i+1]) ALORS
```

```
15              s <- s + 1
16          FIN SI
17          i <- i + 1
18      FIN TANT QUE
19  FIN
```

Calculer le nombre d'affectations, en supposant que chacun des booléens sont vrai ou faux, dans le pire cas, le meilleur cas et le moyen cas ?