

# CORRECTION Examen 1 - Décalé 2023-2024 - CYBER1 (1h30) Algorithmique 1

NOM :

PRÉNOM :

Vous devez respecter les consignes suivantes, sous peine de 0 :

- Lisez le sujet en entier avec attention
- Répondez sur le sujet
- Ne détachez pas les agrafes du sujet
- Écrivez lisiblement vos réponses (si nécessaire en majuscules)
- Ne trichez pas

## 1 Questions (6 points)

1.1 (2 points) Sélectionnez les conditions vraies pour  $A = 8$  et  $B = 5$  :

- ☐  $(A > B)$  et  $((A - 3) > B)$
- ☒  $(A \leq B)$  ou  $((B + 3) \leq A)$
- ☐  $((A - 2) < (B + 2))$  et  $(A > (B + 3))$
- ☒  $((A - 3) \leq B)$  et  $((A / 2) < B)$

1.2 (4 points) Exécutez cet algorithme avec les valeurs  $(x = 3)$   $(y = 2)$  et  $(z = 4)$  en remplissant le tableau, puis donnez les caractéristiques de cet algorithme :

```

algorithme fonction Calc : entier
  parametres locaux
    entier    x, y, z
  debut
    si (y == 1)
      retourne (1)
    sinon
      si ((x % y) == 0)
        retourne (y + Calc(x, (y - 1), z))
      sinon
        retourne (Calc(x, (y - 1), z))
      fin si
    fin si
  fin algorithme fonction Calc
  
```

tour	x	y	z
0	4	3	5
1	4	2	5
2	4	1	5
(3)			

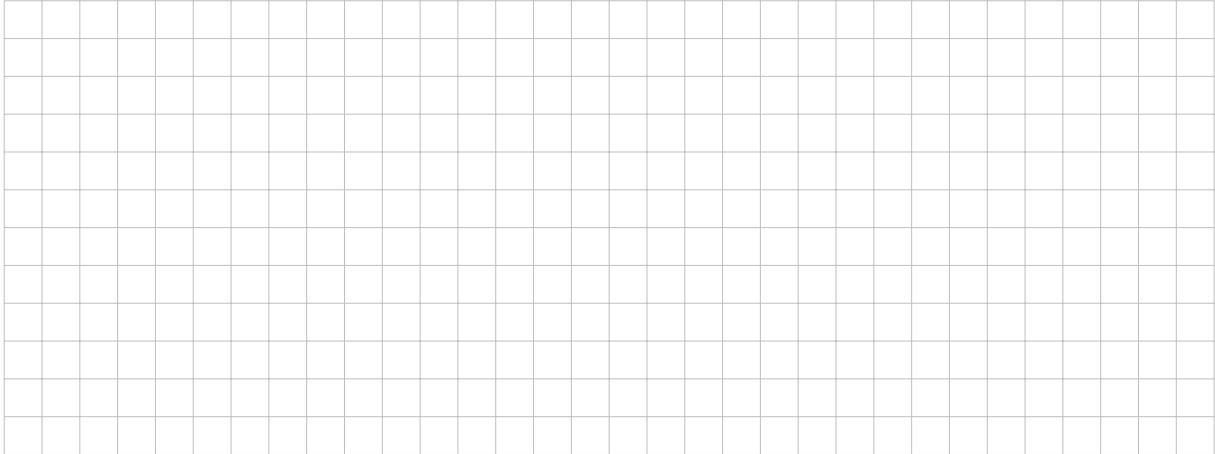
- ☒ Il est récursif
- ☐ Il est même récursif terminal
- ☒ Il s'agit d'une fonction
- ☐ Il s'agit d'une procédure

## 2 Algorithmes (15 points)

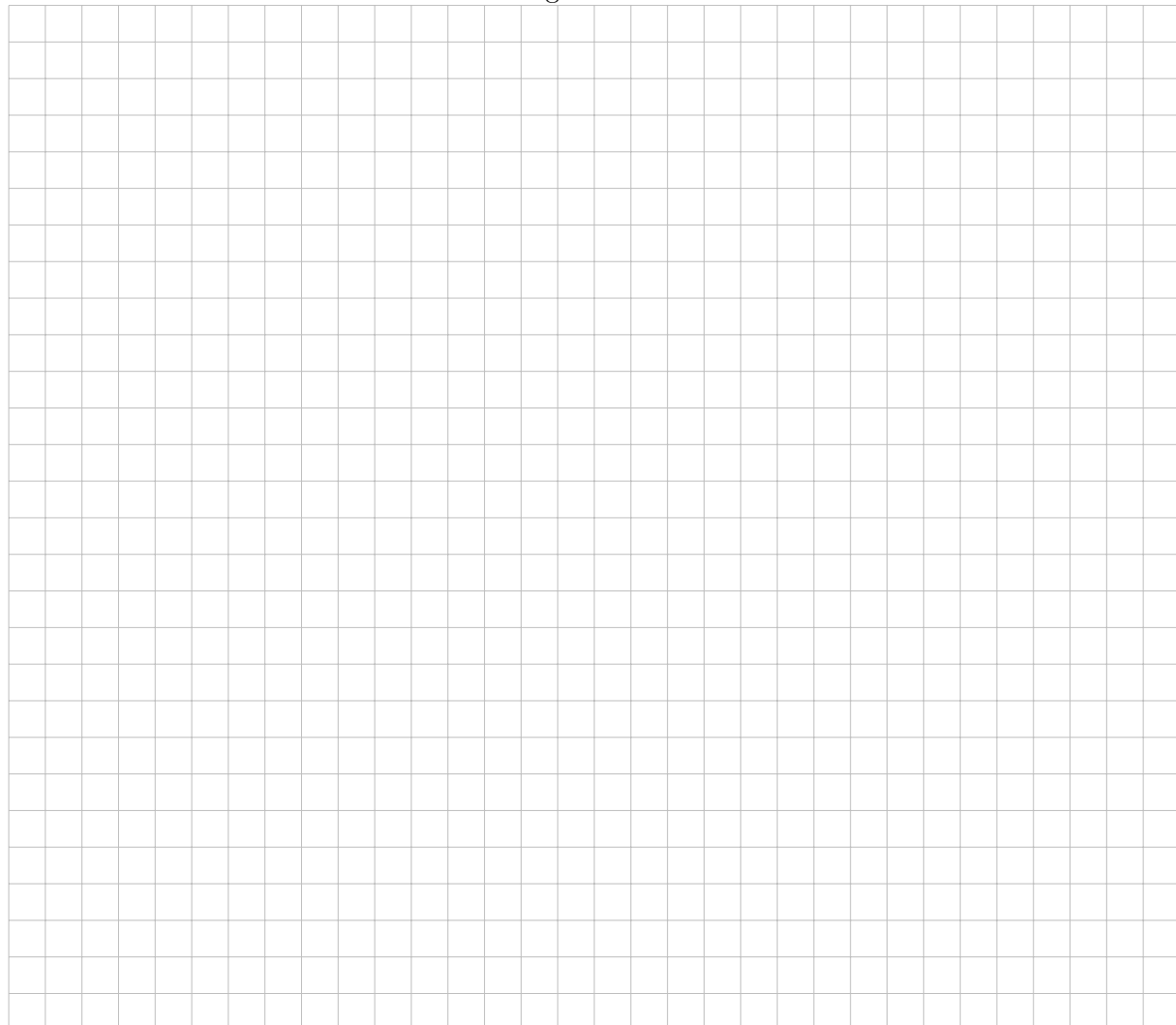
- 2.1 (2 points) Écrivez une fonction « *Factorielle* » récursive terminale. (Vous décrirez d'abord le ou les cas d'arrêts, puis le cas général)

$$n! = \prod_{i=1}^n i = 1 \times 2 \times \dots \times N$$
$$0! = 1$$

Explications du/des cas d'arrêts, puis du cas général :



Algorithme :



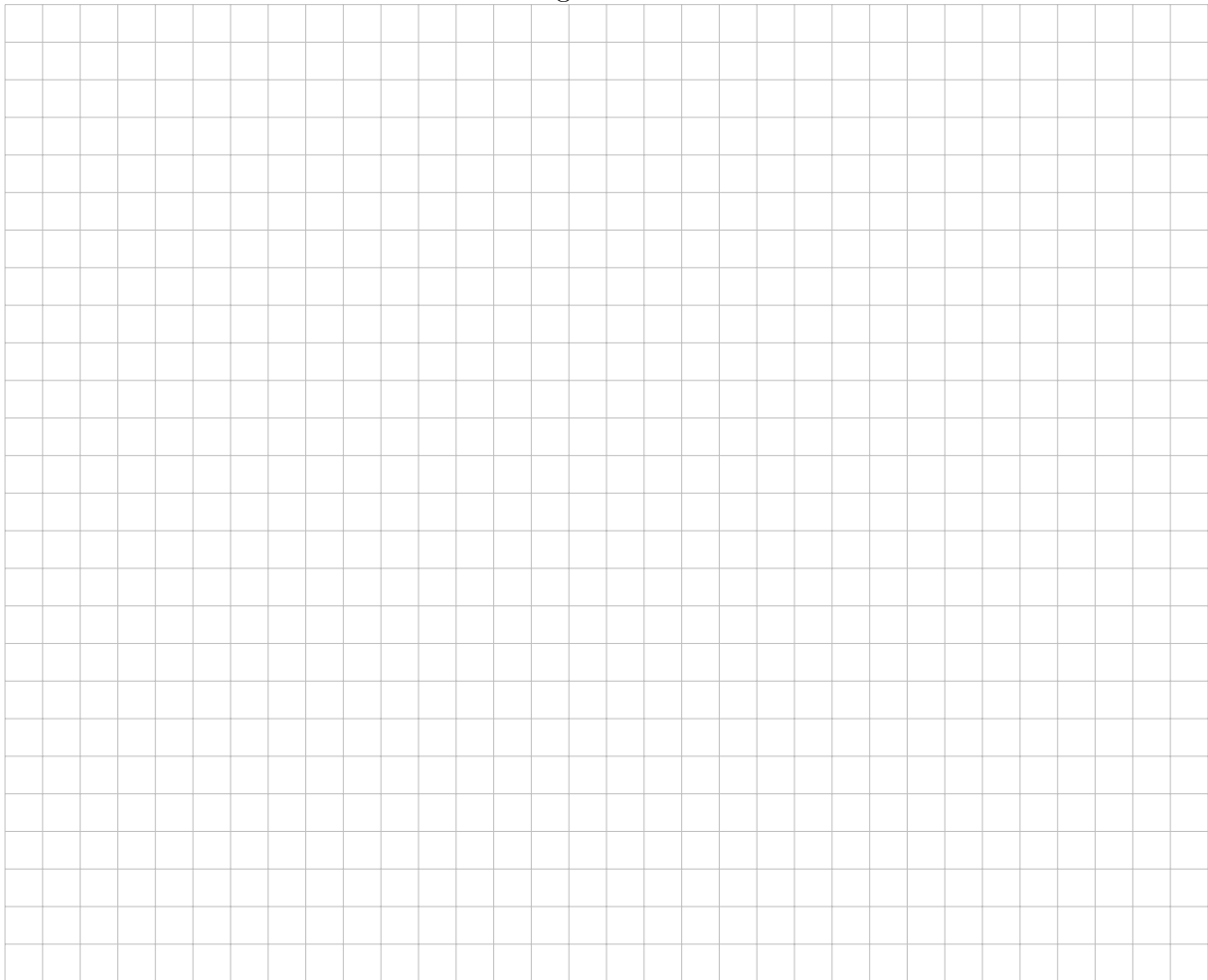
- 2.2 (2 points) Écrivez une fonction « *SommePow* » récursive calculant la somme des N premières puissances d'un nombre M. (Vous décrirez d'abord le ou les cas d'arrêts, puis le cas général)

$$\sum_{i=0}^N M^i = M^0 + M^1 + M^2 + \dots + M^N$$

Explications du/des cas d'arrêts, puis du cas général :

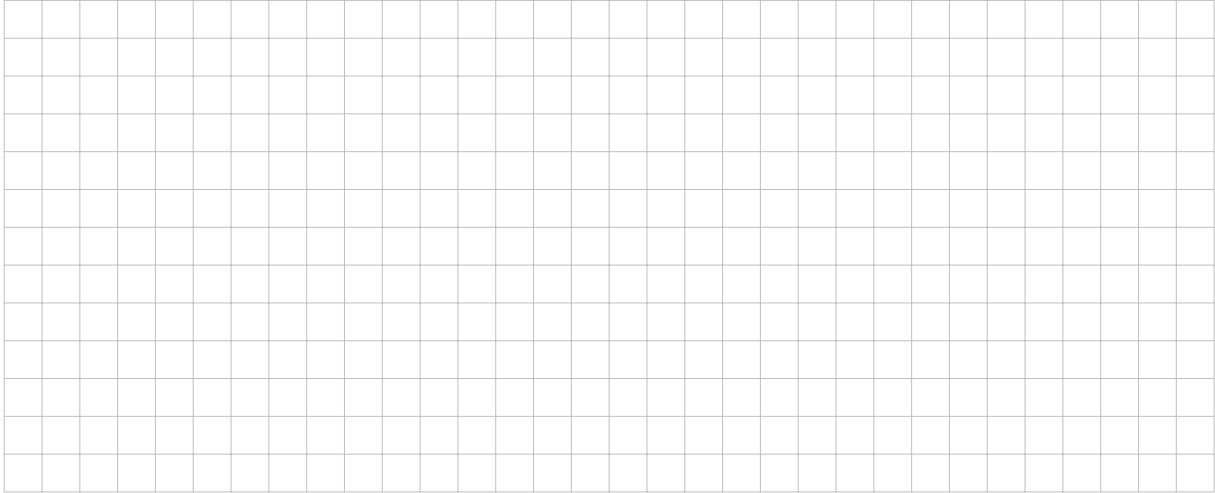


Algorithme :

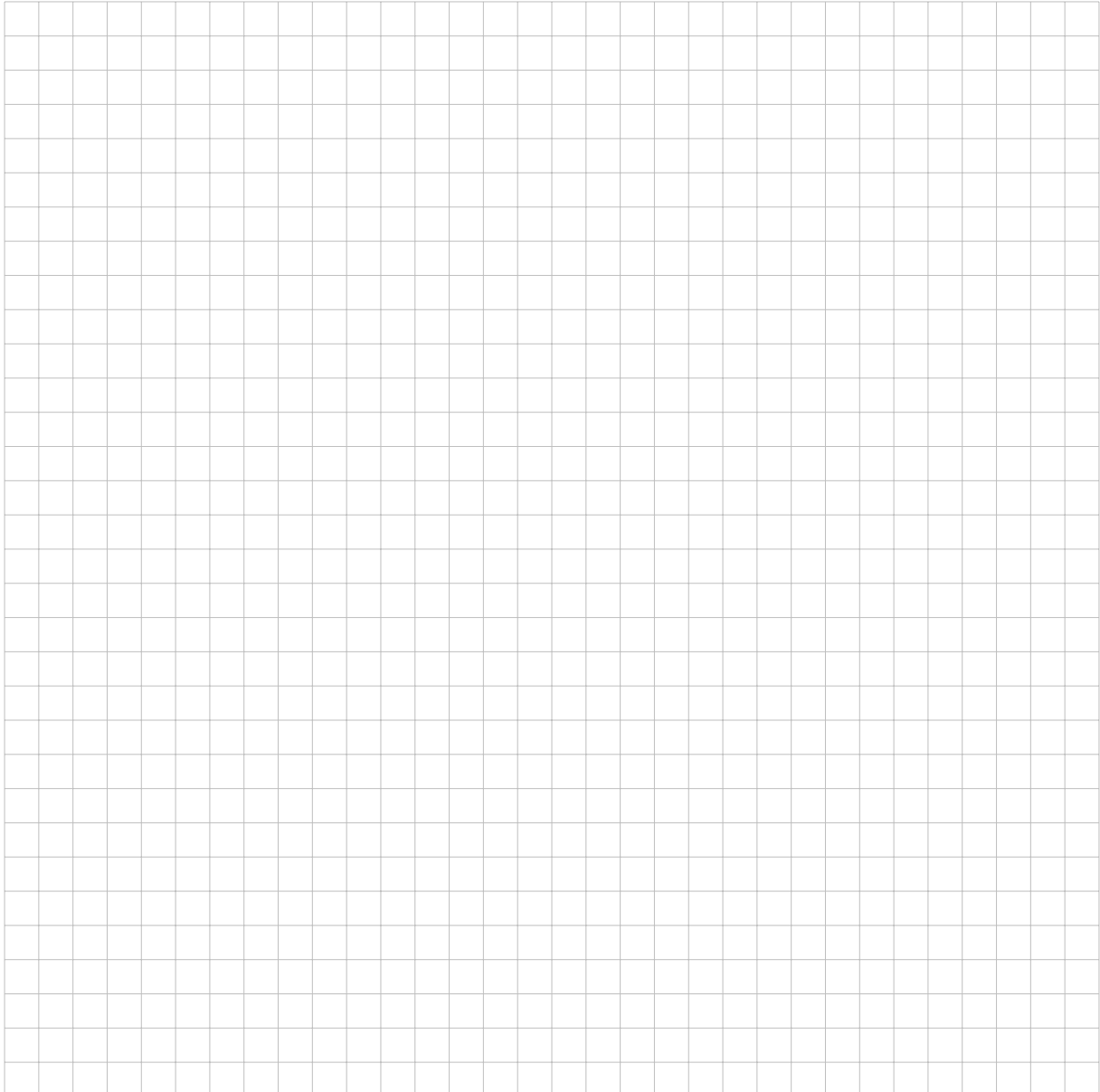


- 2.3 (3 points)** Écrivez une procédure « *NbPairsTab* » itérative affichant la quantité de nombres pairs d'un tableau d'entiers. (Vous décrirez d'abord le fonctionnement général)

Explications du fonctionnement :



Algorithme :



2.4 (3 points) Écrivez une fonction « *TabToInt* » transformant un tableau d'entiers en un unique entier, et l'affichant à l'écran. Chaque case contient un nombre positif entre 10 et 99. Si une case possède un nombre plus petit que 10 ou plus grand que 99, la fonction renverra -1. (Vous décrirez d'abord le fonctionnement général)

42	10	21	32
----	----	----	----

Ce tableau doit devenir 42102132 sous forme d'entier

### Explications du fonctionnement :

Algorithme :

(5 points) Écrivez une fonction « *Swap* » qui échange la position de deux valeurs dans un tableau, puis, écrivez une procédure de tri. (Vous décrirez d'abord le fonctionnement général de l'algorithme de tri)

Swap :

## Explications du fonctionnement :

### Algorithme de tri :

# **SUJET DÉCALÉ ALGORITHMIQUE 1**