

Correction de l'Examen de Mathématiques et Algorithmique

Gemini

13 juin 2025

Remarque Générale

Ce document fournit une correction détaillée des exercices présentés dans les images. Il est important de noter que plusieurs questions ou algorithmes dans l'énoncé original semblent contenir des erreurs, des typos ou des ambiguïtés. Les solutions proposées ci-dessous sont basées sur l'interprétation la plus probable de l'intention de l'examineur, et les incohérences sont signalées le cas échéant.

1 Exercice 4 (Image 1)

1.1 Fonction A

La fonction A est une implémentation de l'algorithme d'Euclide, qui calcule le plus grand diviseur commun (PGCD) de deux entiers n et m.

1.1.1 Question 15

Question : Pour $n=12$ et $m=3$ quelle est la valeur renvoyée par la fonction A ?

Analyse : On calcule le PGCD(12, 3). Comme 12 est un multiple de 3, PGCD(12, 3) = 3. L'algorithme s'exécute comme suit :

- $\text{max} = 12, \text{min} = 3$
- La condition de la boucle **tant que** ($\text{max} \bmod \text{min} \neq 0$) est fausse ($12 \bmod 3 = 0$).
- L'algorithme retourne min , qui est 3.

Réponse : B. 3

1.1.2 Question 16

Question : Pour $n=70$ et $m=15$ quelle est la valeur renvoyée par la fonction A ?

Analyse : On calcule le PGCD(70, 15).

- $\text{max} = 70, \text{min} = 15. r = 70 \bmod 15 = 10.$
- $\text{max} = 15, \text{min} = 10. r = 15 \bmod 10 = 5.$
- $\text{max} = 10, \text{min} = 5. r = 10 \bmod 5 = 0.$
- La boucle s'arrête. Le dernier reste non nul est 5, qui est le PGCD.

Le résultat est 5.

Note : Aucune des options (A. 45, B. 3, C. 15, D. 30) ne correspond à la bonne réponse. La question ou les options proposées contiennent une erreur.

1.1.3 Question 17

Question : Que calcule la fonction A ?

Analyse : Comme expliqué précédemment, la fonction implémente l'algorithme d'Euclide.

Réponse : D. Plus grand diviseur commun

1.1.4 Question 18

Question : Pour $n=100^{99}$ et $m=100^{100}$ quelle est la valeur renvoyée par la fonction A ?

Analyse : On cherche PGCD($100^{99}, 100^{100}$). Sachant que $100^{100} = 100 \times 100^{99}$, 100^{99} est un diviseur de 100^{100} . Le PGCD de deux nombres dont l'un divise l'autre est le plus petit des deux.

Réponse : B. 100^{99}

1.2 Fonction B

L'algorithme B, tel qu'écrit, est ambigu. Pour répondre aux questions, nous devons supposer l'intention du programmeur. L'interprétation la plus logique est que la fonction

B a pour but de compter le nombre d'éléments dans un tableau qui sont des puissances de 2. Cela nécessite que la variable p soit réinitialisée à 1 à chaque itération de la boucle principale.

1.2.1 Question 19

Question : Que retourne la fonction B pour le tableau $T = \{8, 0, 1, 4, 2\}$?

Analyse : En supposant que la fonction compte les puissances de 2 (avec $2^0 = 1$) :

- 8 est une puissance de 2 (2^3).
- 0 n'est pas une puissance de 2.
- 1 est une puissance de 2 (2^0).
- 4 est une puissance de 2 (2^2).
- 2 est une puissance de 2 (2^1).

Il y a 4 nombres qui sont des puissances de 2 dans le tableau.

Réponse : C. 4

1.2.2 Question 20

Question : Que fait la fonction B ?

Analyse : Basé sur l'analyse de la Q19.

Réponse : B. Compte le nombre de puissances de 2 dans T

1.2.3 Question 21

Question : Que retourne la fonction B pour le tableau $T = \{1, 1, 0, 2^4, 2^{10}, 2^{10}, 2^{10}, 2^{12}\}$?

Analyse : Nous comptons les puissances de 2 dans le tableau :

- 1 (présent 2 fois)
- 2^4 (présent 1 fois)
- 2^{10} (présent 3 fois)
- 2^{12} (présent 1 fois)
- 0 n'est pas une puissance de 2.

Le total est $2 + 1 + 3 + 1 = 7$.

Réponse : B. 7

2 Exercice 2 (Image 2)

2.1 Fonction F

La fonction F est une définition réursive classique de la fonction factorielle. $F(n) = n * F(n-1)$ avec $F(0) = 1$. Donc, $F(n) = n!$.

2.1.1 Question 6

Question : Pour $n=4$ quelle est la valeur renvoyée par la fonction F ?

Analyse : $F(4) = 4! = 4 \times 3 \times 2 \times 1 = 24$.

Réponse : A. 24

2.1.2 Question 7

Question : Quelle est la valeur f_n renvoyée par F en fonction de n ?

Analyse : La fonction calcule la factorielle de n.

Réponse : B. $f_n = n!$

2.2 Fonction G

La fonction G est définie par $G(n, m) = n^m + G(n - 1, m)$ avec $G(0, m) = 0$. En développant la récurrence, on obtient : $G(n, m) = n^m + (n - 1)^m + \dots + 1^m + G(0, m) = \sum_{i=1}^n i^m$.

2.2.1 Question 8

Question : Donner la valeur $g_{n,m}$ renvoyée par G en fonction de n et m.

Analyse : D'après l'analyse ci-dessus.

Réponse : B. $g_{n,m} = \sum_{i=1}^n i^m$

3 Exercice 3 (Image 2)

Note importante : Il y a une incohérence majeure dans cet exercice. La fonction M telle qu'elle est écrite ne correspond pas aux questions Q9, Q10 et Q11 et leurs options. La fonction M écrite calcule une valeur liée à la somme des premiers entiers, tandis que les questions suggèrent une fonction liée aux puissances de 3. Nous supposons que la fonction M voulue est la suivante :

```
def M_prime(a : int) -> int :  
    k = 0  
    p = 1  
    while p < a :  
        p = p * 3  
        k = k + 1  
    return k
```

Cette fonction M_prime(a) trouve le plus petit entier k tel que $3^k \geq a$.

3.1 Fonction M (corrigée)

3.1.1 Question 9

Question : Pour a=70 quelle est la valeur renvoyée par la fonction M ?

Analyse : Avec la fonction M_prime, on cherche le plus petit k tel que $3^k \geq 70$. $3^1 = 3, 3^2 = 9, 3^3 = 27, 3^4 = 81$. Le plus petit k est 4.

Réponse : C. 4

3.1.2 Question 10

Question : Que détermine la fonction M ?

Analyse : La fonction M_prime renvoie la plus petite puissance k pour laquelle 3^k est supérieur ou égal à a .

Réponse : A. $\min\{k \in \mathbb{N} / 3^k \geq a\}$

3.1.3 Question 11

Question : Pour $a = 3^{27} + 1$ quelle est la valeur renvoyée par la fonction M ?

Analyse : La question est probablement mal retranscrite. En se basant sur les options, il est très probable que la valeur de a était $a = 3^{99} + 1$. Avec cette hypothèse et la fonction `M_prime`, on cherche le plus petit k tel que $3^k \geq 3^{99} + 1$.

- pour $k = 99$, $3^{99} < 3^{99} + 1$.
- pour $k = 100$, $3^{100} > 3^{99} + 1$.

Le plus petit entier k est donc 100.

Réponse (sous hypothèse) : B. 100

3.2 Fonction N

La fonction N calcule $p = 1 + \sum_{i \in D}$, où D est l'ensemble des diviseurs de n entre 1 et $n/2$. Cela revient à calculer $1 + (\text{somme des diviseurs propres de } n)$. La somme des diviseurs propres de n est $\sigma_1(n) - n$.

3.2.1 Question 12

Question : Que retourne la fonction N pour $n=40$?

Analyse : Les diviseurs de 40 sont $\{1, 2, 4, 5, 8, 10, 20, 40\}$. Les diviseurs propres (excluant 40) sont $\{1, 2, 4, 5, 8, 10, 20\}$. Leur somme est 50. L'algorithme, tel qu'il est écrit, initialise `p=1` puis ajoute les diviseurs. Le résultat serait donc $1 + (1+2+4+5+8+10+20) = 51$. Cependant, 50 est une option et 51 ne l'est pas. Il est probable qu'il y ait une erreur d'initialisation dans le code et que `p` aurait dû être initialisé à 0 pour calculer directement la somme des diviseurs propres, qui est 50.

Réponse (sous hypothèse d'une erreur de code) : B. 50

4 Exercice 5 et questions diverses (Image 3)

4.1 Question 22

Question : Que retourne la fonction H pour le tableau T tel que : T est de taille 1000 et $\forall k \in \{0, \dots, 999\}, T[k] = 1$ si k est impair, 4 si k est pair.

Analyse : La fonction H n'est pas définie. En supposant que H est la fonction B de l'exercice 4 (qui compte les puissances de 2), nous avons :

- Le tableau contient 500 nombres pairs (pour $k = 0, 2, \dots, 998$), donc 500 fois la valeur 4.
- Le tableau contient 500 nombres impairs (pour $k = 1, 3, \dots, 999$), donc 500 fois la valeur 1.
- $4 = 2^2$ est une puissance de 2.
- $1 = 2^0$ est une puissance de 2.

Tous les 1000 éléments du tableau sont des puissances de 2.

Réponse (sous hypothèse $H=B$) : A. 1000

4.2 Question 23

Question : Soient f et g deux suites à valeurs strictement positives. On suppose que : $f(n) = O(g(n))$. Quelle proposition logique cela implique ?

Analyse : La définition de la notation "Grand O" est : il existe une constante $c > 0$ et un rang n_0 à partir duquel $|f(n)| \leq c \cdot |g(n)|$. Comme les suites sont strictement positives, cela devient $f(n) \leq c \cdot g(n)$.

Réponse : B. $(\exists \epsilon > 0)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0) : f(n) < \epsilon \cdot g(n)$ (Note : l'usage de ϵ pour la constante et de l'inégalité stricte ne change pas la nature de la définition de O).

4.3 Question 24

Question : Quelle est la complexité temporelle de l'algorithme suivant ?

Analyse : L'algorithme a deux boucles imbriquées. Nous devons compter le nombre d'exécutions de l'instruction interne 'S = S+1'.

- Pour $i=0$, la boucle interne s'exécute 1 fois ($j=0$).
- Pour $i=1$, la boucle interne s'exécute 2 fois ($j=1, 0$).
- Pour $i=n$, la boucle interne s'exécute $n+1$ fois ($j=n, n-1, \dots, 0$).

Le nombre total d'opérations est $S = 1 + 2 + \dots + (n+1) = \frac{(n+1)(n+2)}{2}$. C'est un polynôme de degré 2 en n .

Réponse : B. $O(n^2)$

4.4 Fonction mystère

Cette fonction utilise une méthode itérative pour converger vers une valeur. Si elle converge, alors $Y \approx X$. En posant $Y = X$ dans l'équation de mise à jour : $X = \frac{2X+A/X^2}{3} \implies 3X = 2X + A/X^2 \implies X = A/X^2 \implies X^3 = A \implies X = \sqrt[3]{A}$. Il s'agit de la méthode de Newton-Raphson pour trouver la racine cubique de A.

4.4.1 Question 25

Question : Quelle relation vérifient X et Y après l'exécution de la boucle ?

Analyse : La boucle se termine lorsque la condition 'Jusqu'à ($\text{abs}(Y-X) \leq \text{epsilon}$)' est satisfaite. C'est la relation qui lie X et Y à la fin de la boucle. Aucune des options ne correspond exactement. L'option B ($|Y-X| > \text{epsilon}$) est la négation de la condition d'arrêt. Il y a probablement une erreur dans les options. La relation correcte est $|Y-X| \leq \epsilon$.

4.4.2 Question 26

Question : Quelle est la relation entre la valeur renvoyée par la fonction mystère et le paramètre A ?

Analyse : Comme démontré ci-dessus, la fonction calcule la racine cubique de A.

Réponse : C. $Y = \sqrt[3]{A}$

5 Exercice 1 (Image 4)

Cet exercice demande de compléter un algorithme qui saisit les données de 20 élèves et trouve celui avec la meilleure moyenne. Les questions sont mal formulées et les numéros de ligne sont incohérents. Nous allons répondre en nous basant sur la logique de l'algorithme.

5.0.1 Algorithme Corrigé

```
// ... (d eclarations)
D but
  // remplir le tableau d'eleves
  Pour_i_de_1_a_20_Faire
    Ecrire(" Eleve_n_", i, ":")
    Ecrire("Nom:")
    Lire(T[i].nom) // Ligne 7
    Ecrire(" Classe:")
    Lire(T[i].classe) // Ligne 9
    Ecrire(" Moyenne:")
    Lire(T[i].moyenne) // Ligne 11
  FinPour
  // Recherche de la moyenne maximale
  moyMax := T[1].moyenne
  iMax := 1
  Pour_i_de_2_a_20_Faire
    Si T[i].moyenne > moyMax alors
      moyMax := T[i].moyenne // Mise a jour
      iMax := i // Mise a jour
    FinSi
  FinPour
  // Affichage
  Ecrire("L'eleve qui a la moyenne maximale est : ",
        T[iMax].nom, T[iMax].classe, T[iMax].moyenne) // Lignes 27-28
Fin
```

5.0.2 Question 1

Question : Pour la ligne 7 quelle est l'instruction correcte : **Analyse :** La ligne 7 doit lire le nom de l'élève. La variable correcte est `T[i].nom`. **Réponse : B. T[i].nom**

5.0.3 Questions 2 et 3

Ces questions sont inexploitable car les numéros de ligne et les options proposées sont incohérents avec le code.

5.0.4 Question 4

Question : Pour la ligne 23 quelle est l'instruction correcte : **Analyse :** En ignorant le numéro de ligne et en se fiant aux options, cette question semble porter sur la mise à jour de la moyenne maximale. **Réponse : B. moyMax <- T[i].moyenne**

5.0.5 Question 5

Question : Pour les lignes 27 et 28 quelles sont les instructions correctes : **Analyse :** Il faut afficher les informations de l'élève ayant la moyenne maximale, qui se trouve à l'indice `iMax`. **Réponse : D. T[iMax].nom T[iMax].classe**