

Ne PAS retourner ces feuilles avant d'en être autorisé!

Merci de poser votre carte CAMIPRO en évidence sur la table.
Vous pouvez déjà compléter et lire les informations ci-dessous:

NOM _____

Prénom _____

Numéro SCIPER _____

Signature _____

BROUILLON : Ecrivez aussi votre NOM-Prénom sur la feuille de brouillon fournie.
Toutes vos réponses doivent être sur cette copie d'examen. Les feuilles de brouillon sont ramassées pour être immédiatement détruites.

Le test écrit commence à : **14h15**

Nous recommandons de finir cet examen vers : **15h35**

Pour commencer l'examen de C++ qui se terminera à : **16h45**

Le contrôle de ICC reste SANS appareil électronique

Vous avez le droit d'avoir tous vos documents **personnels** sous forme papier: dictionnaire, livres, cours, exercices, code, projet, etc...

Total sur 25 points = 16 points pour la partie Quizz et 9 points pour les questions ouvertes
Vous pouvez utiliser un crayon à papier et une gomme

La partie Quizz++ comporte 9 questions : chaque question n'a qu'une seule réponse correcte parmi les 4 réponses proposées. Le nombre de points de chaque question est indiqué sur fond noir, ci-dessous. **Pour les questions ++, un complément de réponse ouverte est exigé** pour obtenir les points de la question : UN COMPLEMENT INCORRECT INVALIDE LA REPONSE DE LA QUESTION DU QUIZZ. Réciproquement, aucun point n'est donné en cas de réponses multiples, de rature, ou de réponse incorrecte au Quizz.

Indiquez vos réponses de la partie Quizz dans ce tableau (pénalité de 1 point en cas d'oubli) :

	Questions du Quizz (16 points)									
	1	2	3	4++	5	6	7	8	9++	
A										A
B										B
C										C
D										D
Barème										
	2	2	2	3	1	1	1	2	2	

QUIZZ ICC Théorie module 1

Dans cet examen, L'opérateur **modulo** calcule le reste dans la division entière de l'opérande de gauche par l'opérande de droite et le caractère / désigne l'opérateur de la division entière sauf indication contraire. En l'absence d'autre précision la fonction **log** désigne la fonction logarithme de base 2.

La base dans laquelle est exprimée un nombre est soit indiquée dans le texte (hexadécimal, décimal, octal, binaire, base N, etc..) ou soit indiqué par un indice ; ex : 101_{10} est un nombre en décimal. S'il n'y a aucune indication de texte ou d'indice le nombre est en base dix.

Coût calcul des opérateurs arithmétiques : Toutes les questions portant sur l'ordre de complexité travaillent sur des nombres correspondant au cadre classique pour lequel les opérateurs arithmétiques suivants ont un coût constant quelle que soit l'amplitude des opérandes : **addition, soustraction, multiplication, division, modulo**.

Coût calcul d'autres opérations : pour cet examen on considère aussi que la fonction **racine carrée** a un coût calcul constant. Par contre, on considère que l'opérateur d'**élévation à la puissance**, noté \wedge , n'est **pas** un opérateur à coût constant.

Question 1 (2pts): binaire, octal et hexadécimal (sans limitation de capacité)

Soit X et Y deux nombres entiers positifs (non-signés) dont la valeur est donnée en **octal** : $X = 2351_8$ et $Y = 4017_8$. Quelle est la valeur du résultat de l'addition $X + Y$ en **hexadécimal** ?

A	$CF8_{16}$
B	6370_{16}
C	6368_{16}
D	$19F0_{16}$

Question 2 (2 pts) : exécution d'un algorithme

Question 3 (2 pts) : ordre de complexité

Algo1
entrée : entier $N > 0$ sortie : aucune (affiche des valeurs)
<pre> i ← N r ← 0 Tant que i > 0 Pour j de 1 à i r ← r + 2 i ← i / 2 // division entière Afficher r </pre>

Estimer le nombre d'instructions exécutées par l'algorithme Algo1 en fonction de N lorsque N tend vers l'infini.

Simplifier l'expression obtenue en sachant que $\sum_{k=0}^{\infty} \frac{1}{2^k} = 2$

Quelle valeur est affichée pour N valant 4 :

En déduire l'ordre de complexité de Algo1 :

A	0
B	8
C	14
D	Aucune des autres valeurs

A	$O(N \log(N))$ mais pas $O(N)$
B	$O(N^2)$ mais pas $O(N \log(N))$
C	$O(\log(N))$ mais pas $O(1)$
D	$O(N)$ mais pas $O(\log(N))$

Question 4++ (3pts): algorithme avec 3 variables en virgule flottante IEEE 754 simple précision ayant une précision $\varepsilon = 2^{-23}$

Algo2
entrée : 3 nombres en virgule flottante IEEE 754 simple précision : start, stop, step tels que : $\text{stop} > \text{start} > 0$ et $0 < \text{step} < (\text{stop} - \text{start})$ sortie : <i>aucune</i> , réalise des actions d'affichage
$x \leftarrow \text{start}$ Tant que $x < \text{stop}$ Afficher x Afficher passage à la ligne $x \leftarrow x + \text{step}$

Indiquer la réponse correcte pour l'appel **Algo2(2.²³, 2.²⁵, 1.)** :

A	Cet appel affiche x jusqu'au moment où x est égal à stop ce qui fait quitter la boucle et terminer l'algorithme
B	Cet appel affiche x jusqu'au moment où $x > \text{stop}$
C	Cet appel n'entre jamais dans la boucle et termine l'algorithme sans affichage
D	L'algorithme ne se termine pas car il entre dans une boucle infinie pour cet appel

⇒ **Complément exigé pour valider votre réponse ci-dessus :** **Justifier votre réponse**

Question 5 (1 pt): représentation des données

Un système de représentation des couleurs choisit de représenter une couleur en indiquant une quantité pour chacune des trois couleurs suivantes : Rouge, Vert et Bleu.

Sachant que la quantité de chacune des composantes (rouge, verte et bleue) est représentée avec deux octets, combien de couleurs distinctes est-il possible de définir avec cette représentation ?

A	6^{16}
B	256^6
C	48^2
D	Aucune des autres réponses

Question 6 (1 pt): Quelle est la complexité de **Algo3** en fonction de n ?

Algo3
entrée : entier n strictement positif sortie : <i>non précisée</i>
Si $n=1$ Sortir 1 Sortir $n * \text{Algo3}(n-1)$

A	$O(n!)$ mais pas $O(2^n)$
B	$O(\log(n))$ mais pas $O(1)$
C	$O(n)$ mais pas $O(\log(n))$
D	$O(n^2)$ mais pas $O(n)$

Question 7 (1pt) : Représentation des entiers en **complément à 2** sur une capacité de **4 bits**

A	Le domaine couvert est $[-7, 8]$
B	Le domaine couvert est $[-8, 7]$
C	Le domaine couvert est $[-8, 8]$
D	Le domaine couvert est $[-7, 7]$

Question 8 (2 pts):

Dans la représentation de la question précédente, le résultat de l'addition : $1011_2 + 1111_2$

A	produit une violation du domaine couvert
B	donne -6
C	donne 6
D	donne une autre valeur dans le domaine couvert (différente de 6 et de -6)

Question 9++ (2 pts) : Quelle affirmation est VRAIE

A	Le problème de l'arrêt appartient à la classe NP
B	Le problème du sac à dos est dans P grâce au tri des objets selon le critère V/P
C	Le problème du tri appartient à la classe NP
D	Il n'existe pas de surjection de l'ensemble des entiers vers l'ensemble des rationnels

⇒ **Complément exigé pour valider votre réponse ci-dessus : Justifier votre réponse**

Question Ouverte 1 : (5 pts) Soit une liste **L** contenant **N** entiers avec $N > 0$

1.1) Ecrire le pseudocode de l'algorithme **Tous_Differents** qui renvoie **VRAI** si **tous les éléments de L sont différents entre eux**.

Par exemple l'algorithme renvoie VRAI pour $L = \{4, 9, 12\}$ tandis qu'il renvoie FAUX pour $L = \{4, 9, 12, 11, 9, 22\}$. L'algorithme renvoie VRAI s'il n'y a qu'un seul élément dans L. Formulé d'une autre manière, on peut dire que cet algorithme doit renvoyer FAUX dès qu'il trouve deux éléments égaux.

On demande une version itérative, sans récursivité, pour cette question

Tous_Differents	<i>(faisable en 5 lignes)</i>
entrée : entier $N > 0$, Liste L contenant N entiers	
sortie : booléen VRAI si tous les éléments sont différents entre eux (sinon renvoie FAUX)	

1.2) Quelle est la complexité de votre algorithme en fonction de **N** ?

Justifier votre réponse :

1.3) fournir une **version récursive de votre algorithme** (qu'on appellera **TDR**). Faire comme dans le cours sur le tri fusion pour la syntaxe du passage d'une sous-liste lors d'un appel récursif: après la taille de la sous-liste, indiquer le nom de la liste **L** avec, entre parenthèses, les indices du premier et du dernier élément transmis pour la sous-liste.

Exemple : voici un appel récursif qui transmet la sous-liste contenant les deux derniers éléments de L : **TDR(2, L(N-1, N))**

TDR	<i>(faisable en 8 lignes)</i>
entrée : entier N > 0 , Liste L contenant N entiers	
sortie : booléen VRAI si tous les éléments sont différents entre eux (sinon renvoie FAUX)	

1.4) Quelle est la complexité de votre algorithme en fonction de N ?

Justifier votre réponse :

Question Ouverte 2 : (4 pts)

Soit l'algorithme mystère **Algo4**

Algo4
entrée : entier $N > 0$, Liste L contenant N valeurs entières 0 ou 1, et entier P tel que $0 < P \leq N$ sortie : à déterminer
$i \leftarrow 1$ $k \leftarrow P - 1$ Sortir Algo5(N, L, i, k)

Avec **Algo5** :

Algo5
entrée : $N > 0$, Liste L contenant N valeurs entières 0 ou 1, entier $i > 0$, entier k sortie : à déterminer
Si $i = N$ Sortir $L(i) * 2^k$ Sortir $L(i) * 2^k + \text{Algo5}(N, L, i+1, k-1)$

2.1) Indiquer la valeur de la **sortie de Algo4** lorsque que cet algorithme est appelé avec N valant 4, $L = \{1, 0, 1, 1\}$ et P valant 3. Préciser chacun des appels (récursifs) de **Algo5**, en particulier la valeur renvoyée à chaque appel.

Remarque importante pour cet exercice: l'accès aux éléments de L se fait de gauche à droite, c'est-à-dire que $L(1)$ vaut 1, $L(2)$ vaut 0, $L(3)$ vaut 1 et $L(4)$ vaut 1 .

2.2) Quel type de calcul est réalisé par ces deux algorithmes, c'est-à-dire : quelle est la nature du résultat en fonctions des paramètres ?

2.3) En déduire la valeur renvoyée par **Algo4** pour les mêmes paramètres sauf P valant 2 puis 1.

⇒ Pour P valant 2 :

⇒ Pour P valant 1 :

2.4) proposer un algorithme unique itératif sortant le même résultat que **Algo4** sans appeler cet algorithme.

Algo6	<i>(faisable en 6 lignes)</i>
entrée : entier N > 0, Liste L contenant N valeurs entières 0 ou 1, et entier P tel que $0 < P \leq N$	
sortie : <i>la même que Algo4</i>	