

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 puis détruites.

Le test écrit commence à :

14h15

Retourner les feuilles avec la dernière page face à vous à :

15h30

*les contrôles écrits ICC sont SANS document autorisé,
ni appareil électronique*

Total sur 20 points = 12 points pour la partie Quizz et 8 points pour les questions ouvertes

La partie Quizz (QCM) comporte 12 questions : chaque question n'a qu'une seule réponse correcte parmi les 4 réponses proposées. Chaque réponse correcte donne 1 point. Il n'y a pas de pénalité en cas de mauvaise réponse. Aucun point n'est donné en cas de réponses multiples, de rature, ou de réponse incorrecte. Vous pouvez utiliser un crayon à papier et une gomme.

Indiquez vos réponses à la partie Quizz dans **le tableau en bas de cette page**.

La partie « question ouverte » comporte 2 questions. Chaque question rapporte 4 points.

	Questions du Quizz													
	1	2	3	4	5	6	7	8	9	10	11	12		
A														A
B														B
C														C
D														D

QUIZZ

Notation: on utilise dans ce quizz la virgule pour séparer les puissances positives des puissances négatives de la base dans la notation positionnelle des nombres.

Nombres entiers sur 32 bits.

Question 1 : cette question travaille avec des nombres non-signés. Quel est le résultat en hexadécimal de l'addition des deux nombres hexadécimaux: $1234_{16} + 9986_{16}$

- A BABA
- B BOBO
- C DADA
- D ABBA

Question 2 : soit le nombre hexadécimal $FFF4_{16}$; on suppose que son motif binaire est la représentation d'un nombre décimal dans la représentation du *complément à deux* sur 16 bits. Ce nombre décimal est :

- A -4
 - B -12
 - C -10
 - D 12
-

Nombres à virgule sur 8 bits.

Question 3 : quelle est le nombre décimal représentée par le motif binaire 10101001 dans la représentation *non-signée* à virgule fixe avec 4 bits de partie entière et 4 bits de partie fractionnaire :

- A $21,125_{10}$
- B $10,125_{10}$
- C $9,5675_{10}$
- D $10,5625_{10}$

Question 4 : On s'intéresse à la représentation du nombre décimal $17,5_{10}$ avec la représentation à virgule flottante avec 4 bits pour l'exposant de la base et 4 bits pour la mantisse.

- A Ce nombre est exactement représenté par le motif binaire 01000001
 - B Ce nombre est exactement représenté par le motif binaire 01000010
 - C Ce nombre est approché par troncation avec le motif binaire 01000001 en faisant une erreur absolue de $0,5_{10}$
 - D Ce nombre est approché par troncation avec le motif binaire 01000001 en faisant une erreur absolue de $0,25_{10}$
-

Etant donnée une liste *non-triée* d'entiers relatifs, on s'intéresse au problème de trouver la plus grande distance entre deux valeurs de cette liste (cette distance est définie par la valeur absolue de la différence de 2 valeurs). Par exemple, pour la liste $\{16, -1, -7, 21\}$, la réponse est 28.

Question 5 : Le meilleur algorithme capable de résoudre ce problème pour une liste *non-triée* (de taille connue N) a pour ordre de complexité :

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

Etant donnée une liste *triée* d'entier relatifs, on veut résoudre le même problème que pour la question précédente.

Question 6 : Le meilleur algorithme capable de résoudre ce problème pour une liste *triée* (de taille connue N) a pour ordre de complexité :

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

Soit l'algorithme **Algo_X** qui reçoit une liste **L** en entrée. Un élément d'une liste **L** peut être accédé avec la notation **L(i)**, avec l'indice **i** compris entre **1** et **Taille(L)**. L'ordre de complexité de **Taille(L)** est **$O(1)$** .

Remarque : l'opérateur de division produit le quotient de la division entière

Algo_X
entrée : liste L avec au moins 2 éléments sortie : liste S de même taille que la liste L
$t \leftarrow \text{Taille}(L)$ $x \leftarrow t/2$ Pour i allant de x à t $S(i - x + 1) \leftarrow L(i)$ Pour i allant de 1 à x-1 $S(t - i + 1) \leftarrow L(i)$ Sortir : S

Question 7 : que fait cet algorithme ?

- A Les éléments de la liste S sont dans l'ordre inverse de ceux de la liste L
- B La liste S commence par les éléments de L compris entre x et t, puis ceux compris entre 1 et x-1
- C La liste S commence par les éléments de L compris entre x et t, puis ceux compris entre 1 et x-1 mais dans l'ordre inverse (d'abord x-1 ensuite x-2 etc...)
- D La liste S commence par les éléments de L compris entre x et t mais dans l'ordre inverse (d'abord t, puis t-1, etc..), puis ceux compris entre 1 et x-1 mais dans l'ordre inverse (d'abord x-1, puis x-2 etc...)

Question 8 : Quelle est son ordre de complexité ?

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

Soit l'algorithme récursif **Algo_R(L,min,max)** qui reçoit une liste **L** en entrée ainsi que deux valeurs d'indices **min** et **max**. Un élément d'une liste **L** peut être accédé avec la notation **L(i)**, avec l'indice **i** compris entre **1** et **Taille(L)**. L'ordre de complexité de **Taille(L)** est **O(1)**.

Remarque : l'opérateur de division produit le quotient de la division entière

Algo_R
entrée : liste non-vide L d'entiers positifs, min, max sortie : un entier x
Si min = max Si (L(min) est impair) Sortir : 1 Sinon Sortir : 0 Sinon Sortir : Algo_R(L, min, min+(max-min)/2) + Algo_R(L, ??? , max)

Question 9 : cet algorithme compte le nombre d'entiers impairs dans la liste **L** lorsqu'il est appelé avec : **Algo_R(L, 1, Taille(L))**. Indiquer quelle expression doit remplacer « ??? » dans le pseudocode pour que cette tâche soit correctement effectuée.

- A (max – min)/2 + 1
- B min + (max - min)/2 + 1
- C min + (max - min)/2
- D max - (max – min)/2 + 1

Question 10 : Quelle est son ordre de complexité ?

- A $O(\log(N))$ mais pas $O(1)$
- B $O(N^2)$ mais pas $O(N)$
- C $O(N)$ mais pas $O(\log(N))$
- D $O(2^N)$ mais pas $O(N^3)$

Question 11 : Indiquer la proposition correcte

- A On sait avec certitude à l'heure actuelle que si un problème est dans la classe NP, alors il est aussi dans la classe P.
- B Si un problème est dans la classe P, alors il n'est pas dans la classe NP.
- C On sait avec certitude à l'heure actuelle qu'il existe un problème dans la classe NP qui n'est pas dans la classe P.
- D Si un problème est dans la classe P, alors il est aussi dans la classe NP.

Question 12 : Appelons PNP un problème connu pour être dans NP. Par rapport au problème de « l'arrêt de programmes » (*halting problem*) vu en cours, que peut-on dire ?

- A PNP est plus difficile que « l'arrêt de programme »
- B Que toute solution de « l'arrêt de programme » est facilement vérifiable
- C Qu'on peut calculer une solution de PNP et de « l'arrêt de programmes » en temps non-polynomial
- D Que PNP est décidable tandis que « l'arrêt de programme » est indécidable

Questions Ouvertes

Question 1 : algorithme de conversion.

1) Soit l'algorithme **Algo_1** qui reçoit une liste **L** en entrée et produit un entier positif **x** en sortie. Un élément d'une liste **L** peut être accédé avec la notation **L(i)**, avec l'indice **i** compris entre **1** et **Taille(L)**. L'ordre de complexité de **Taille(L)** est **O(1)**

La liste **L** contient seulement des 0 et des 1 et représente un nombre binaire non-signé avec le poids fort de la base2 au début de la liste (**L(1)**) et ainsi de suite jusqu'au poids faible en fin de liste **L(Taille(L))**.

Ecrire **Algo_1** pour qu'il calcule la valeur entière **x** en base dix correspondant au nombre binaire représenté par la liste **L**

Algo_1 : conversion du binaire vers un nombre en base dix
Entrée : une liste non-vide L contenant des 0 et des 1
Sortie : un entier x

2.1) On veut écrire l'**Algo_2** pour qu'il calcule la liste **O** des chiffres en base 8 correspondant au nombre binaire représenté par une liste **L** du même type que pour la question 1). Par construction, chaque élément de la liste **O** contient un chiffre compris entre 0 et 7 : **O(1)** contient le chiffre associé au poids fort de la base 8, etc...

Tout d'abord justifier quelle est la taille minimum de la liste **O**, notée **Taille_O**, à partir de **Taille(L)** (on pose que l'opérateur de la division produit le quotient de la division entière, et on appelle **modulo** l'opérateur qui donne le reste dans la division entière):

.....

.....

.....

.....

.....

.....

.....

2) Quelle est l'**erreur absolue maximum** qu'il est possible de faire sur un nombre réel x appartenant au domaine couvert mais n'étant pas exactement représenté par cette représentation en virgule fixe.

.....
.....
.....
.....
.....
.....
.....
.....

3) On se propose de calculer la moyenne géométrique M d'une liste de n nombres réels $\{x_1, x_2, \dots, x_n\}$ exactement représentés avec cette représentation en virgule fixe.

Par définition, $M = \sqrt[n]{x_1 x_2 \dots x_n}$

Pour effectuer ce calcul on suppose qu'on dispose des opérateurs arithmétique habituels et des algorithmes suivants (optionnels, selon votre choix):

- pow(x,n)** : calcule la puissance $n^{\text{ième}}$ de x
- root(x,n)** : calcule la racine $n^{\text{ième}}$ de x
- log(x)** : calcule le logarithme de x ,
- exp(x)** : calcule l'exponentielle de x (la fonction inverse de $\log(x)$)

En supposant que les nombres $\{x_1, x_2, \dots, x_n\}$ sont tous supérieurs à 1, quel **risque** prend-on en appliquant directement la formule fournie pour M ?

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

4) Transformer la formule pour réduire le risque identifié dans le contexte de la question 3)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Ne rien écrire sur cette page,

Rappel : avez-vous complété le tableau en p1 ?

Présenter cette page sur le dessus dans les 2 cas suivants :

- 1) vous avez fini avant 15h30*
- 2) les copies sont ramassées*