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	 	
Dránom		
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 11 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	
Α													Α
В													В
С													С
D													D

QUIZZ

Question 1: Quelle est la représentation binaire en complément à deux sur 7 bits de -10₁₀

- A 0001010
- B 1001010
- C 1110110
- D 1000101

Question 2: Sachant qu'une image RGB est codée avec 256 niveaux d'intensité par couleur et a 3 canaux de couleur par pixel, combien faut-il d'octets pour encoder une image de taille 100x100 ?

- A 7 680 000
- B 240 000
- C 30 000
- D 5 760 000

Question 3: Quelle est le meilleur ordre de complexité pour un algorithme de recherche de N éléments quelconques dans une liste non-triée de taille N.

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

Question 5: Supposons connu un algorithme de complexité en O(Nlog(N)) qui permet de trouver la solution d'un problème de décision appelé « TRUC » qui travaille sur des données de taille N. Que peuton en déduire ?

- A « TRUC » n'est pas dans NP
- B « TRUC » n'est pas dans P
- C « TRUC » est dans NP
- D rien du tout

Question 6: Pour deux entier strictement positifs a et b, l'algorithme de la multiplication égyptienne de a par b revient à :

- A Décomposer les deux opérandes en produit de facteurs premiers pour recombiner les puissances des facteurs premiers et effectuer les produits de ces termes
- B Effectuer une décomposition d'un opérande en une somme de puissances de deux pour réduire le nombre d'additions à calculer sur des multiples de deux de l'autre opérande
- C Effectuer une recherche dichotomique du plus petit opérande parmi les diviseurs du plus grand opérande
- D'abord effectuer une recherche du plus grand diviseur commun avant de l'élever au carré et de poursuivre le produit avec les autres facteurs plus simples

Question 7: L'algorithme suivant doit calculer le PGCD de deux entiers naturel non nuls x et y fournis en entrée. Choisir la combinaison correcte d'instructions manquantes pour INSTRUCTION1 et INSTRUCTION2:

```
PGCD

Entrée : deux entiers strictement positifs x et y

Sortie : leur PGCD

Tant que x ≠ y

Si x > y

INSTRUCTION1

Sinon

INSTRUCTION2

sortir: x
```

Le caractère / désigne la division entière.

```
A INSTRUCTION1= y \leftarrow y - x et INSTRUCTION2= x \leftarrow x - y
B INSTRUCTION1= x \leftarrow x / y et INSTRUCTION2= y \leftarrow y / x
C INSTRUCTION1= x \leftarrow x - y et INSTRUCTION2= y \leftarrow y - x
D INSTRUCTION1= y \leftarrow x - y et INSTRUCTION2= x \leftarrow y - x
```

Question 8: soit l'algorithme suivant qui doit chercher la valeur maximum contenue dans une matrice A de N lignes et M colonnes (dans le cas général M≠N). L'accès à un élément d'une matrice A se fait comme en math à l'aide de deux indices qui commencent à 1 et varient jusqu'au nombre de ligne/ de colonne. On indique **d'abord l'indice de ligne suivi par l'indice de colonne** : A(1,2) désigne l'élément de la première ligne et qui est dans la deuxième colonne de cette ligne.

```
entrées :
deux entiers strictement positif N et M
une matrice A de N lignes et M colonnes contenant des entiers signés
sortie : la valeur entière maximum contenue dans la matrice A

max ← A(1,1)

Pour i de 1 à N

Pour j de 1 à M

Si A(i, j) > max

max ← A(i,j)

sortir: max
```

Quel est l'ordre de complexité de cet algorithme avec la notation de Landau ?

- A $O(N^M)$
- B $O(N^2)$
- C O(NM)
- D O(N+M)

Question 9: Si on suppose maintenant que les valeurs de la matrice A fournie à l'algorithme **MatMax** sont triées dans l'ordre croissant sur chaque ligne mais qu'il n'y a pas d'ordre particulier entre les lignes. Quelle est la proposition correcte ?

- A L'algorithme MatMax ci-dessus s'exécute avec une complexité de O(N)
- B L'algorithme MatMax ci-dessus s'exécute avec une complexité de O(M)
- C L'algorithme MatMax peut être modifié pour obtenir une complexité qui ne dépend plus de M
- D L'algorithme MatMax peut être modifié pour obtenir une complexité qui ne dépend plus de N

Question 10: Soit une représentation en virgule flottante en binaire utilisant 5 bits pour la mantisse et 3 bits pour l'exposant. On suppose qu'on utilise toujours la forme normalisée de cette représentation. Que peut-on dire sur l'erreur relative ?

- A Elle vaut au maximum 0.5 sur le domaine couvert
- B Elle vaut au maximum 1/5 sur le domaine couvert
- C Elle vaut au maximum 1/32 sur le domaine couvert
- D Elle vaut 1/32 pour la valeur min de l'exposant puis double pour chaque incrémentation de l'exposant

Question 11 : Soit Algo_X un algorithme recevant en entrée un entier strictement positif N et une liste L de taille N contenant des entiers positifs. Le caractère / désigne la division entière.

```
Algo_X
entrée : entier strictement positif N, liste L de taille N contenant des entiers positifs sortie : ???

z \leftarrow 0
Pour i de 1 à N
a \leftarrow 0
x \leftarrow N
Tant que (x > 0)
a \leftarrow L[x] * 2
x \leftarrow x / 2
z \leftarrow z + a
sortir: z
```

Quel est l'ordre de complexité de cet algorithme avec la notation de Landau ?

- A O(N log N) mais pas O(N)
- B O(N) mais pas O(log(N))
- C O(log N) mais pas O(1)
- D O(N²) mais pas O(Nlog(N))

Question 12 : Quel est le résultat de Algo X ci-dessus ?

- A le double de la somme des éléments de la liste L
- B la somme des éléments de la liste L élevés à la puissance N
- C la somme des éléments de la liste L multipliés par 2^N
- D le double du premier élément de la liste multiplié par N

Questions Ouvertes

Question 1: Liste des sommes partielles d'une liste L de N éléments

Soit L une liste non-vide de N entiers. On définit une liste P des sommes partielles, de même taille N que la liste L, de la manière suivante : chaque élément P(i) est égal à la somme des éléments de la liste L compris entre 1 et i (1 et i étant inclus dans cette somme):

$$P(i) = \sum_{j=1}^{i} L(j)$$
, pour i de 1 à N.

Par exemple, si L = $\{4, 10, 9, 2\}$, alors P = $\{4, 14, 23, 25\}$.

Somme_partie	le
	N>O, liste L d'entiers de taille N, Liste P de taille N contenant des zéros
	es sommes partielles de L
) Justifier l'or	lre de complexité de votre algorithme en fonction de N avec la notation de Landau e
().	
on ajoute à	l'algorithme un paramètre supplémentaire de largeur (Width) sous forme d'un entie
rictement po	sitif W, inférieur ou égal à N. Ce paramètre W indique le nombre maximum de termes
	somme partielle, à partir du terme d'indice i et en remontant vers le début de la liste
	i une liste S dont l'élément S(i) est la somme des éléments de L compris entre L(i) et L(
	mple, si W vaut 1, $S(i) = L(i)$. Si W vaut 2, $S(3)=L(3)+L(2)$. Si W est plus grand que
	it s'assurer que les indices utilisés sont toujours strictement positifs. Par exemple, si V
_	limite les termes pour $S(3)$ avec $S(3) = L(3) + L(2) + L(1)$.
aut 3, alois oi	illilite les terriles pour 5(5) avec 5(5) - L(5)+L(2)+L(1).
roposez un al	gorithme qui calcule les sommes partielles avec largeur W. Vous pouvez ré-utiliser votr
lgorithme de	a question 1 mais cela n'est pas obligatoire.
Somme_partie	_
	N>0 et 0 <w≤n, contenant="" d'entiers="" de="" des="" l="" liste="" n="" n,="" s="" taille="" th="" zéros<=""></w≤n,>
sortie : liste S d	es sommes partielles de L avec une largeur de W

Question 2: Fusion de deux listes triées

Soit un algorithme FusionListes qui reçoit en entrée un entier strictement positif N, deux listes A et B de N éléments triés dans l'ordre croissant et une liste C de taille 2N contenant des zéros. Le but de l'algorithme est fusionner les éléments des listes A et B dans la liste C de telle façon que la liste C soit dans l'ordre croissant. On conserve les doublons, c'est-à-dire que si un élément de la liste A se trouve aussi dans la liste B alors il apparaîtra deux fois dans la liste C.

- 1) Soit les listes suivantes pour A et B, indiquer quelle est la liste C attendue en sortie :
- 1.1) A={-2, 5, 11, 49}, B={8, 11, 20, 30}, C=
- 1.2) A={66, 70, 75, 88}, B={3, 20, 33, 60}, C=
- 2) proposez un algorithme pour résoudre ce problème

FusionListes					
entrée : entier N>0, listes A et B de taille N contenant des valeurs entières triées dans l'ordre croissant, liste C					
de taille 2N contenant des zéros					
sortie : liste C contenant la fusion de A et B dans l'ordre croissant					

3) Quel est l'ordre de complexité de votre algorithme avec la notation de Landau ? Argumentez votre choix.

Vous pouvez utiliser cette page

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