

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**

Retourner les feuilles avec la page 10 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
Vous pouvez utiliser un crayon à papier et une gomme

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. Aucun point n'est donné en cas de réponses multiples, de rature, ou de réponse incorrecte.

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

La partie « question ouverte » comporte 2 questions = 4 points + 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

Unités de taille utilisées dans cet examen

Les puissances de dix utilisent une seule lettre majuscule comme par exemple :
 $10^3 = \text{K (kilo)}$, $10^6 = \text{M (méga)}$, $10^9 = \text{G (giga)}$, $10^{12} = \text{T (téra)}$

Dans cet examen, nous utilisons la lettre majuscule 'B', comme **Byte**, pour désigner un **octet**.
La lettre minuscule 'b' désigne un **bit**.

Opérateurs logiques

and (et logique), or (ou inclusif), xor (ou exclusif), not (négation logique)

Syntaxe des instructions assembleur

Attention : certaines questions utilisent seulement un sous-ensemble de ces instructions

Chargement d'un registre : Il est possible de remplacer r2 par une constante :

<code>charge r1, r2</code>	range dans r1 la valeur stockée dans r2
<code>mcharge r1, r2</code>	range dans r1 la valeur stockée en mémoire à l'adresse r2

Calcul : Il est possible de remplacer r3 par une constante :

<code>somme r1, r2, r3</code>	calcule r2 + r3 et range le résultat dans r1
<code>soustrait r1, r2, r3</code>	calcule r2 - r3 et range le résultat dans r1
<code>multiplie r1, r2, r3</code>	calcule r2 * r3 et range le résultat dans r1
<code>divise r1, r2, r3</code>	calcule le quotient entier de r2/r3 et le range dans r1

Branchement : Il est possible de remplacer r2 par une constante :

<code>continue_diff r1, r2, i</code>	Si r1 est différent de r2 alors continue à l'adresse i Sinon passe à l'instruction suivante.
<code>continue_egal r1, r2, i</code>	Si r1 est égal à r2 alors continue à l'adresse i Sinon passe à l'instruction suivante.
<code>continue_ppe r1, r2, i</code>	Si r1 ≤ r2 alors continue à l'adresse i Sinon passe à l'instruction suivante.
<code>continue_pp r1, r2, i</code>	Si r1 < r2 alors continue à l'adresse i Sinon passe à l'instruction suivante.
<code>continue_pge r1, r2, i</code>	Si r1 ≥ r2 alors continue à l'adresse i Sinon passe à l'instruction suivante.
<code>continue_pg r1, r2, i</code>	Si r1 > r2 alors continue à l'adresse i Sinon passe à l'instruction suivante.
<code>continue i</code>	branchement inconditionnel à l'adresse i

Divers : arrêt et fin du programme avec l'instruction **stop**

QUIZZ

Soit le programme assembleur (incomplet) qui doit mémoriser dans un registre le résultat du **produit scalaire de deux vecteurs à trois dimensions**. On suppose que les registres **r1** à **r3** contiennent les valeurs des coordonnées du premier vecteur, et les registres **r4** à **r6** contiennent les valeurs des coordonnées du second vecteur.

```
1 : charge    r7, 0
2 : multiplie r8, r1, r4
3 : somme     r7, r7, r8
4 : ???
5 : somme     r7, r7, r8
6 : multiplie r8, r3, r6
7 : somme     r7, r7, r8
8 : stop
```

Voir la p2 pour les conventions utilisées pour les instructions.

Question 1 : Comment faut-il compléter le programme à la ligne 4 pour produire le résultat recherché ?

- A multiplie r7, r2, r5
- B multiplie r8, r2, r5
- C multiplie r2, r5, r8
- D multiplie r8, r3, r6

Question 2 : Une mémoire cache contient trois blocs. Chaque bloc contient deux mots. Un bloc est toujours aligné sur une adresse paire d'un mot de la mémoire centrale, c'est-à-dire qu'un bloc contient les deux mots consécutifs d'adresses $\langle 2n, 2n+1 \rangle$ avec n entier naturel.

La stratégie de remplacement d'un bloc lorsque que le cache est plein est la stratégie FIFO, c'est-à-dire « first-in, first-out », ce qui se traduit par : le premier chargé est le premier enlevé, indépendamment de son utilisation.

Indiquer quelle séquence de chargement de mots produit le moins de défaut de cache parmi les 4 séquences suivantes :

- A 0, 16, 33, 65, 0, 16, 33, 65
- B 1, 33, 65, 64, 32, 0, 1, 65
- C 0, 1, 2, 3, 4, 5, 6, 7
- D 1, 2, 5, 1, 2, 5, 6, 2

Question 3 : Une expérience de physique produit une quantité de données de 1 TB sur une durée de une seconde. On dispose de disques durs d'une capacité de 0,5TB avec une bande passante de 100 MB/s pour chaque disque. Combien de disques doit-on utiliser en parallèle pour pouvoir enregistrer cette quantité de données en temps réel, c'est-à-dire sur la durée de une seconde.

- A 2
- B 1'000
- C 2'000
- D 10'000

Question 4 : Quelle est la fonction d'une politique d'ordonnancement dans la lecture d'un disque dur ?

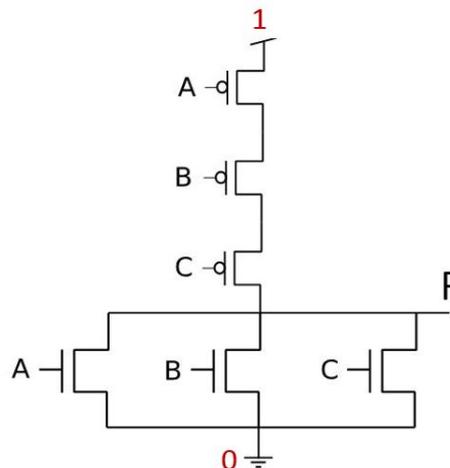
- A Optimiser la quantité de données stockée sur un disque
- B Accélérer la lecture de chaque requête indépendamment des autres
- C Accélérer la lecture d'un ensemble de requêtes
- D Exploiter une base de données relationnelle

Question 5 : Soit un nombre binaire sur trois bits compris entre 000_2 et 111_2 . On pose que les 3 variables booléennes A, B et C sont vraies lorsque le bit correspondant respectivement à 2^2 , 2^1 et 2^0 est à 1. Elles sont fausses sinon. Par exemple, pour le nombre binaire 011_2 la variable A est fautive, tandis que B et C sont vraies.

Quelle est la fonction logique F qui est vraie si et seulement si le nombre binaire est une puissance entière de deux, en fonction de A, B et C ? (voir notations en page 2)

- A A or B or C
- B A and B and C
- C (A and not (B or C)) or (B and not (A or C)) or (C and not (A or B))
- D (A or not (B and C)) and (B or not (A and C)) and (C or not (A or B))

Question 6 : Quelle la fonction logique réalisée par la sortie F de ce circuit à partir des trois entrées A, B et C ?



(voir notations en page 2)

- A $F = A \text{ and } B \text{ and } C$
- B $F = A \text{ or } B \text{ or } C$
- C $F = \text{not } (A \text{ or } B \text{ or } C)$
- D $F = \text{not } (A \text{ and } B \text{ and } C)$

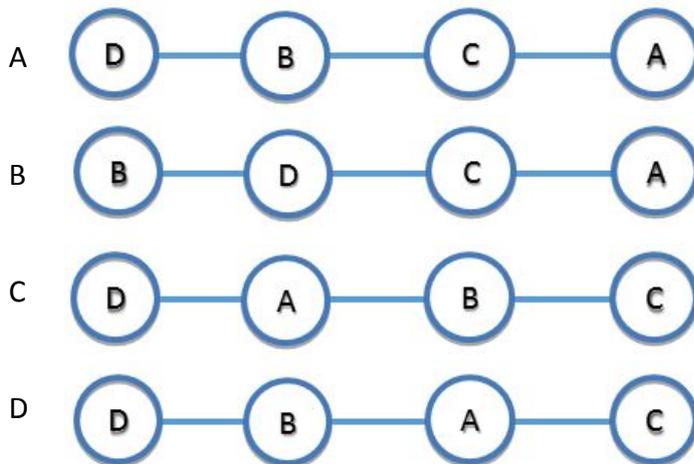
Question 7 : Un ordinateur dispose d'un processeur fonctionnant à 2 GHz (un tic de l'horloge toutes les 0.5ns) et d'un disque dur de bande passante 400 MB/s. Quel est l'ordre de grandeur du nombre d'additions que peut faire le processeur pendant le temps nécessaire pour charger 1MB de données du disque dur vers la mémoire centrale ? (On suppose que l'addition s'effectue en un tic d'horloge)

- A 100'000
- B 5'000
- C 1'000'000'000
- D 5'000'000

Question 8 : On donne la table de routage suivante

D			C		
dest.	dir.	dist.	dest.	dir.	dist.
A	B	2	A	A	1
B	B	1	B	A	2
C	B	3	D	A	3

Quel est le réseau correspondant à cette table de routage ?



Question 9 : Soit le programme assembleur suivant (voir p2 pour les conventions) :

```

1: charge r4, r2
2: charge r3, 0
3: somme r3, r3, r1
4: somme r4, r4, -1
5: ?
6: ?
7: stop

```

Quelles doivent être les lignes 5 et 6 afin d'obtenir dans un registre le résultat de la multiplication des deux valeurs entières contenues dans les registre r1 r2?

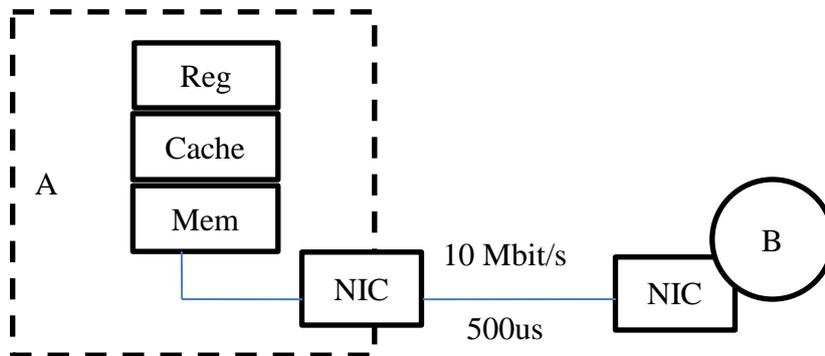
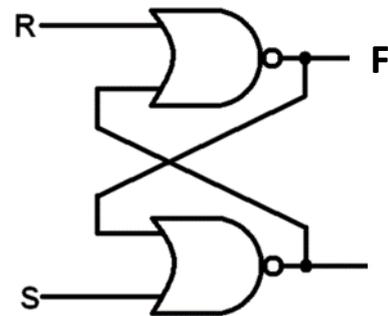
- A 5: continue_egal r4, 0, 3
6: continue 7
- B 5: continue_egal r4, 0, 7
6: continue 3
- C 5: continue_egal r3, 0, 7
6: continue 3
- D 5: continue_egal r3, 0, 3
6: continue 7

Question 10: Le circuit suivant sert à mémoriser un bit d'information que l'on peut lire sur la sortie F.

Quelle est la fonction logique obtenue à la sortie F de ce circuit à partir de R, S et de F ?

(voir notations en page 2)

- A $F = \text{not}(\text{not}(F \text{ or } S) \text{ or } R)$
- B $F = \text{not}(F \text{ or } S) \text{ or } R$
- C $F = \text{not}(\text{not}(F \text{ and } S) \text{ or } R)$
- D $F = \text{not}(F \text{ and } S) \text{ or } R$



L'ordinateur A est connecté à l'ordinateur B par l'intermédiaire de sa carte d'interface réseau (NIC) avec une bande passante de réseau de 10Mb/s et une latence de 500 μ s. Par ailleurs, voici les coûts d'une opération d'écriture entre les éléments de l'ordinateur : d'un registre vers la mémoire cache : 10ns, du cache vers la mémoire centrale : 100ns, de la mémoire centrale jusqu'à la carte NIC : 5 μ s.

Question 11 : Sachant qu'il faut 510,11 μ s pour transférer un octet d'un registre de l'ordinateur A jusqu'à l'entrée du NIC de l'ordinateur B, quelle est la latence causée par la carte NIC de l'ordinateur A dans cette opération de transfert ?

- A 5 μ s
- B 10ns
- C 100ns
- D La carte NIC ne cause aucune latence supplémentaire

Question 12 : En considérant seulement la bande passante entre les 2 cartes NIC, quelle durée faut-il pour transférer 200 MB de données ?

- A 150 seconds
- B 20 seconds
- C 160 seconds
- D 200 seconds

Questions Ouvertes

Question 1 : Assembleur (4 pts) Soit le code assembleur suivant :
(voir conventions en p2)

```
1 : multiplie      r3, r1, r1
2 : continue_egal r3, r2, 12
3 : continue_pg    r3, r2, 10
4 : divise         r3, r2, r1
5 : multiplie      r4, r3, r1
6 : soustrait      r3, r2, r4
7 : continue_egal r3, 0, 12
8 : somme           r1, r1, 1
9 : continue       1
10: charge         r4, 1
11: continue       13
12: charge         r4, 0
13: stop
```

a) Que vaut r4 lorsque le programme ci-dessus se termine, si au départ la valeur binaire de r1 vaut 00000010 et celle de r2 vaut 00001101 ? Justifiez votre réponse en donnant les valeurs intermédiaires des registres.

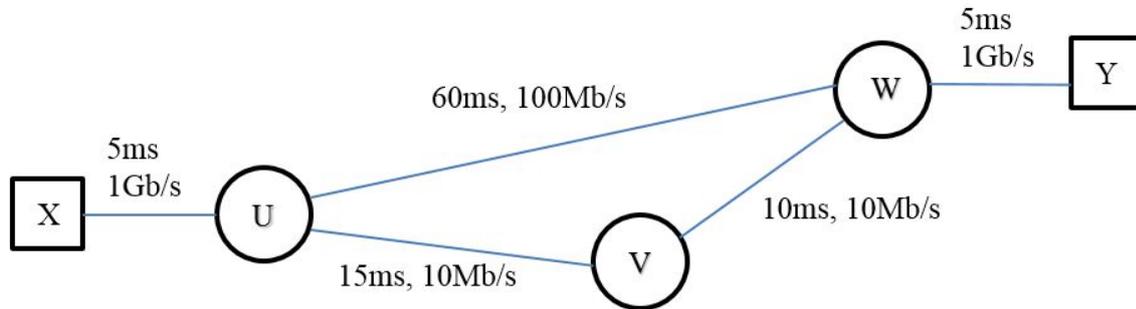
b) Ecrivez le pseudocode de l'algorithme correspondant au programme ci-dessus

c) En une phrase, que fait cet algorithme (mathématiquement) ?

d) Quelle est l'ordre de complexité de cet algorithme en fonction de r2 ? Pourquoi ?

Question 2 : Routage (4 pts)

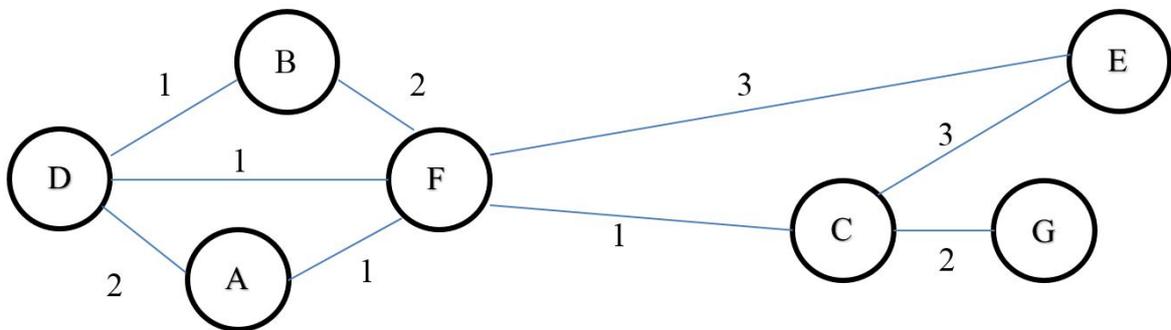
a. Indiquer la latence minimum entre les ordinateurs X et Y en précisant comment vous la déterminez:



b. Quelle route est préférable pour transférer le plus rapidement possible un fichier de 15 GB de X à Y ? Pourquoi ?

c. Quelle est la bande passante maximum réalisable sur cette route (de X à Y) ?

d. dessiner les tables de routage des nœuds F et E. Remarquez que dans cet exercice on indique une distance à coté de chaque lien. Par exemple la distance de A à D est 2.



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*