

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 :

16h15

Nous recommandons de consacrer 1h20 à l'examen de C++ et 1h40 à l'examen théorique

Les deux copies d'examens sont ramassées à :

19h15

***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, notes manuscrites, etc...

Vous pouvez utiliser un crayon à papier et une gomme

Total sur 50 points = 35 points pour la partie Quizz++ et 15 points pour les questions ouvertes

La partie Quizz++ comporte 12 questions : chaque question n'a qu'une seule réponse correcte parmi les 4 réponses proposées. **Pour les questions ++**, le nombre de points est plus important car **un complément de réponse ouverte est exigé** pour obtenir les points de la question. UN COMPLEMENT INCORRECT INVALIDE LA REPONSE DE LA QUESTION. Réciproquement, aucun point n'est donné en cas de réponses multiples, de rature, ou de réponse incorrecte au Quizz. Le barème est indiqué sur l'avant-dernière ligne, sur fond noir, ci-dessous.

Indiquez vos réponses à la partie Quizz dans ce tableau :

	Questions du Quizz (35 pts)													
	1++	2	3	4	5	6	7	8++	9	10++	11++	12		
A														A
B														B
C														C
D														D
	5	2	2	2	2	3	2	4	2	5	4	2		
	Barème													

Quelques conventions

Dans cet examen, L'opérateur **modulo**, noté **mod**, 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. 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 10.

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**. Par contre, on considère que l'opérateur d'**élévation à la puissance**, noté \wedge en pseudocode, n'est *pas* un opérateur à coût constant car il demande une boucle de multiplications.

Unités de taille utilisées dans cet examen

Les puissances de dix utilisent une seule lettre majuscule comme par exemple :

$10^3 = K$ (kilo), $10^6 = M$ (méga), $10^9 = G$ (giga), $10^{12} = T$ (téra)

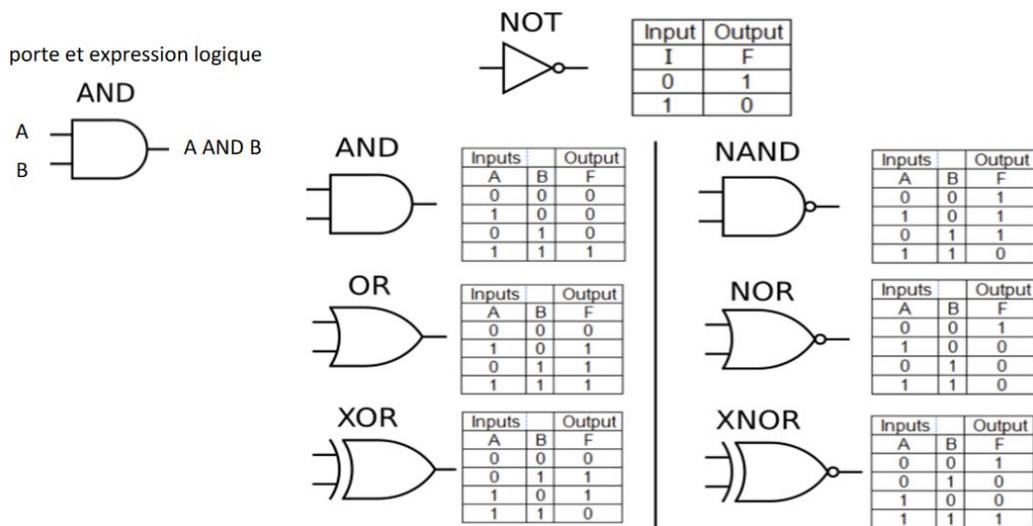
Dans cet examen, nous utilisons la lettre majuscule 'B', comme **Byte**, pour désigner un **octet**. La lettre minuscule 'b' désigne un **bit**.

Opérateurs logiques pour le pseudocode

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

Portes logiques et circuits à base de transistors

Soit x une variable logique, la notation \bar{x} signifie la négation de x : si $x=0$ alors $\bar{x}=1$ et vice versa.



Formules trigonométriques

$$\sin(u) \sin(v) = 0.5(\cos(u - v) - \cos(u + v))$$

$$\cos(u) \sin(v) = 0.5(\sin(u + v) - \sin(u - v))$$

Question 1++ (5 pts): codage mystère avec l'algorithme de **Huffman**

Un ensemble de caractères a été codé avec l'algorithme de Huffman. On ne connaît pas le nombre de caractères différents ni le nombre d'apparitions pour chaque caractère.

Voici deux indices pour trouver le nombre de caractères différents qui ont été codés:

1. les codes obtenus avec l'algorithme de Huffman sont tous d'une longueur comprise entre **1 et 4 au maximum**.
2. On connaît déjà le code de deux caractères : il s'agit des codes **0 et 10**.

Quel est le **nombre maximum** de caractères différents **supplémentaires** (c'est-à-dire, en plus des deux caractères qui ont déjà leur code) qu'il a été possible de coder avec l'algorithme de Huffman ?

QUIZZ++

A	Aucune des autres réponses
B	2
C	4
D	8

Complément exigé : donner les **codes** supplémentaires

--

Question 2 (2 pts) : soit deux représentations non signées pour représenter des nombres à virgule positifs sur 8 bits :

(a) virgule fixe avec **4 bits** pour la partie entière et **4 bits** pour la partie fractionnaire

(b) virgule flottante avec **3 bits** pour l'exposant (sans biais) et **5 bits** pour la mantisse.

On utilise la forme *dénormalisée* quand la valeur de l'exposant est **zéro**.

Quelle proposition est FAUSSE ?

A	Le domaine couvert par (b) est plus grand que celui couvert par (a)
B	Il existe un intervalle de valeurs pour lequel l'erreur relative faite avec (a) est plus petite ou égale que celle faite avec (b)
C	La valeur 65_{10} est représentée exactement avec la représentation (b)
D	L'ensemble des nombres exactement représentés dans l'intervalle $[0, 2[$ est le même dans les deux représentations

Question 3 (2 pts) : suite de la question2 ; quelle proposition est VRAIE ?

A	La précision de (a) est 2^{-4} et celle de (b) est 2^{-5} (pour l'intervalle couvert par la forme normalisé)
B	L'erreur absolue faite avec (a) est au maximum de 2^{-4} et l'erreur absolue maximum de (b) double quand on passe à la puissance de 2 suivante (pour l'intervalle couvert par la forme normalisé).
C	L'erreur relative de (b) est aussi majorée par 2^{-5} dans l'intervalle couvert par la forme dénormalisée
D	N valeurs sont exactement représentées entre deux puissances de deux consécutives (la puissance supérieure de 2 étant exclue de ce nombre), avec N valant 32 pour la représentation (b) et 16 pour la représentation (a)

Algo1	
entrée : entier $N > 2$, Liste L contenant N entiers	
sortie : aucune (affiche des valeurs)	
$x \leftarrow 0$ $y \leftarrow 0$ $k \leftarrow 0$ Pour i de 2 à N-1 Si $L(i) < L(i-1)$ Si $L(i) < L(i+1)$ $k \leftarrow k+1$ Si $x = 0$ $\text{min_min} \leftarrow L(i)$ Afficher min_min $x \leftarrow 1$ Sinon Tant que $(L(i) < \text{min_min})$ $\text{min_min} \leftarrow L(i)$ Afficher min_min Si $y = 0$ $\text{max_min} \leftarrow L(i)$ Afficher max_min $y \leftarrow 1$ Sinon Tant que $(L(i) > \text{max_min})$ $\text{max_min} \leftarrow L(i)$ Afficher max_min Afficher k	

Question 4 (2pts): Quel **affichage** est produit par l'exécution de **Algo1** lorsqu'il est appelé avec N valant 10 et la liste suivante L : {20 , 25 , 25 , 20 , 15 , 20 , 12 , 20 , 17 , 20 }

Remarque : l'affichage ajoute des espaces après chaque valeur affichée

A	0
B	15 15 12 17 3
C	25 25 2
D	aucune des autres réponses

Question 5 (2 pts) : Quelle est la complexité de **Algo1** en fonction de N ?

Remarque : **Afficher** a un coût constant

A	Aucune des autres réponses
B	$O(\log(N))$ mais pas $O(1)$
C	$O(N^2)$ mais pas $O(N)$
D	$O(N^3)$ mais pas $O(N^2)$

Codage mystère avec l'algorithme de **Huffman** (bis)

Un ensemble de caractères contenant N caractères différents et $N > 3$ a été codé avec l'algorithme de Huffman. Cependant on ne connaît pas les nombres d'apparitions des caractères.

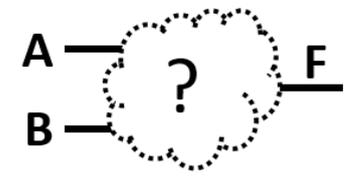
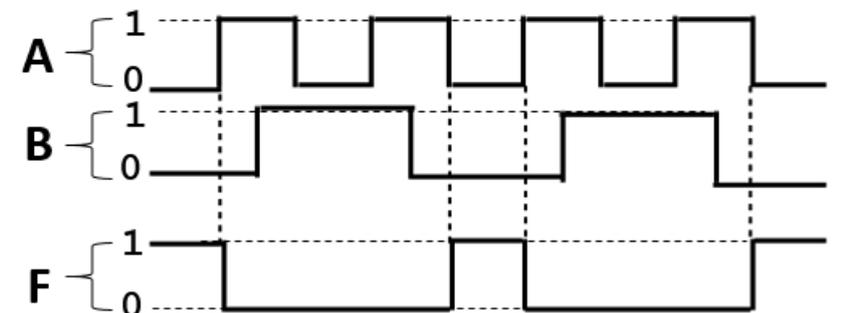
Question 6 (3 pts): Quelle est la **plus grande longueur de code possible** pour ces N caractères ?

A	N
B	N-1
C	N/2 (division entière)
D	Aucune des autres réponses

Question 7 (2 pts) : Pour **quelle configuration d'apparitions** des N caractères différents obtient-on la plus grande longueur de code possible ?

A	Aucune des autres réponses
B	Tous les nombres d'apparitions des caractères sont différentes puissances de 2 et leur somme n'est pas une puissance de 2
C	Toutes les fréquences d'apparitions des caractères sont des puissances négatives de 2 et l'entropie de l'ensemble des caractères est strictement inférieure à la performance du code obtenu avec Huffman.
D	Toutes les fréquences d'apparitions sont égales

Question 8++ (4 pts): Trouver une porte logique à partir d'un chronogramme

<p>Un chronogramme est une représentation temporelle. Il montre les changements de valeur entre 0 et 1 des entrées A et B et de la sortie F d'une porte logique. A partir du chronogramme ci-dessous il est possible de déterminer la table de vérité d'une porte logique.</p>	
<p>Par exemple, on voit sur la gauche du chronogramme que la sortie F vaut toujours 1 quand les deux entrées A et B valent 0.</p>	

QUIZZ++

Quelle est la porte logique ?

A	Aucune des autres réponses
B	NAND
C	NOR
D	XNOR

Complément exigé : donner la table de vérité

--

Question 9 (2 pts): ordre de complexité

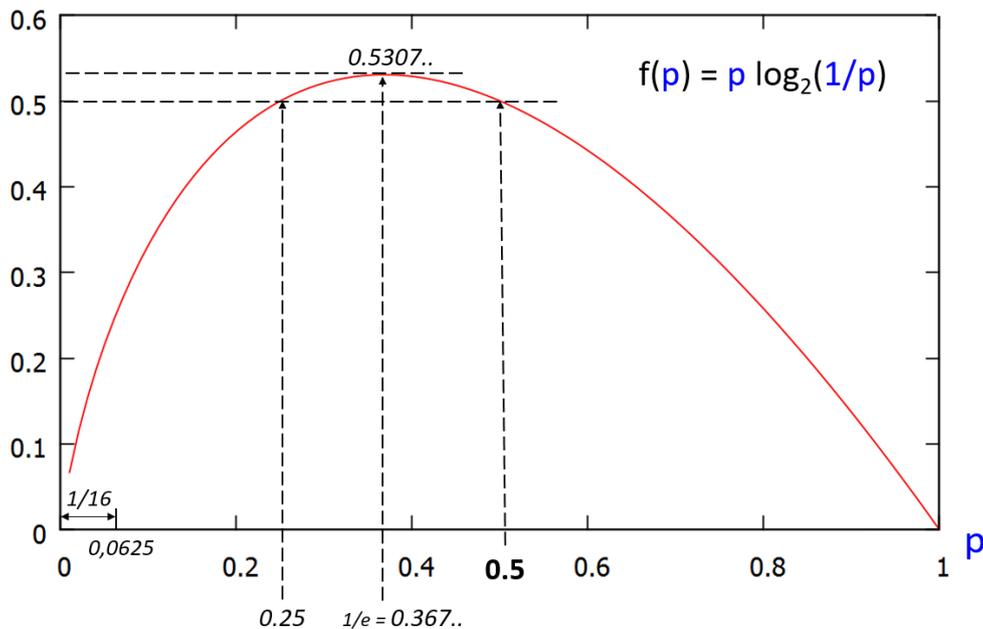
Soit une liste L contenant N entiers avec $N > 0$. On suppose également que tous les éléments de la liste L sont différents entre eux. Quelle est la complexité de l'algorithme qui reçoit L en entrée et qui produit en sortie la liste de toutes les permutations possibles de L ?

Exemple : pour $L = \{1, 2, 3\}$, l'algorithme construit la liste contenant l'ensemble des listes suivantes: $\{ \{1, 2, 3\}, \{1, 3, 2\}, \{2, 1, 3\}, \{2, 3, 1\}, \{3, 2, 1\}, \{3, 1, 2\} \}$

A	$O(n^2)$ mais pas $O(n)$
B	$O(n^3)$ mais pas $O(n^2)$
C	$O(n!)$ mais pas $O(2^n)$
D	$O(n)$ mais pas $O(\log(n))$

Question 10++ (5 pts) : Classification d'entropies à partir de variations de probabilités

Utiliser le graphe ci-dessous pour classer les entropies des séquences de 16 lettres suivantes :		Il n'est pas possible de calculer explicitement toutes les entropies car la fonction \log_2 n'est pas à disposition. Il est cependant possible de calculer facilement l'une des 4 entropies puis d'en déduire comment les autres entropies varient les unes par rapport aux autres. Il n'est pas nécessaire de quantifier la variation ; il suffit de détecter si l'entropie reste constante, augmente ou décroît. Pour vous aider, le dessin ci-dessous montre la quantité $1/16$ qui correspond à un changement d'une unité dans un groupe de lettres identiques.
N° séqu	séquence	
S1	AAAARRRRRGGGGHHHHH	
S2	AAAARRRRRGGGGHHHH	
S3	AAAARRRRRGGHHHHHHH	
S4	AAARRRRRGGHHHHHHH	



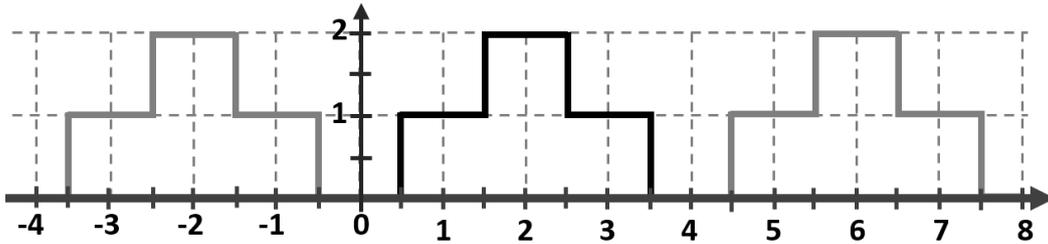
QUIZZ++

A	$H(S2) = H(S4) > H(S1) > H(S3)$
B	$H(S2) > H(S1) > H(S3) > H(S4)$
C	$H(S1) > H(S2) > H(S3) > H(S4)$
D	Aucune des autres réponses

Complément exigé : donner l'entropie maximum

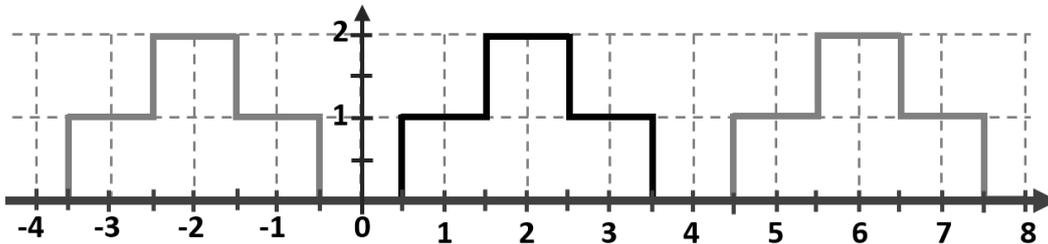
Question Ouverte 1 : (7 pts) Filtre à moyenne mobile

Soit le signal périodique suivant de période $T = 4$ s :

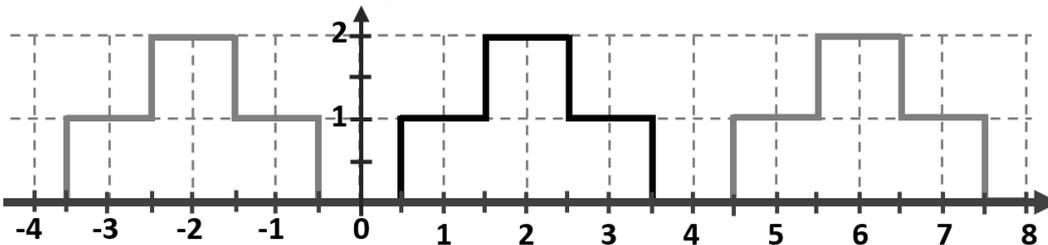


Les 3 questions suivantes demandent d'effectuer un filtrage à moyenne mobile. Pour **chaque** réponse on demande de justifier la forme du signal filtré en indiquant quel **type d'équation** permet de le représenter (ex : le signal initial est construit par morceau, chaque morceau est de la forme $y = \text{Constante}$). De plus préciser **les minima et maxima** sur une période (s'il y en a).

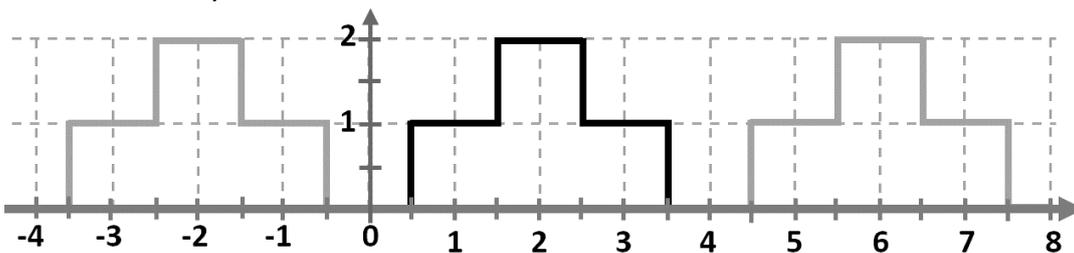
1.1) Dessiner sur le dessin ci-dessous le résultat du filtrage après le passage d'un filtre à moyenne mobile de période d'intégration $T_c = 4$ s.



1.2) Dessiner maintenant sur le dessin ci-dessous le résultat du filtrage après le passage d'un filtre à moyenne mobile de période d'intégration $T_c = 1$ s.



1.3) Recopier votre dessin en réponse à la question précédente (**noté R**) sur le dessin ci-dessous, puis dessiner par-dessus le résultat du filtrage de **R** avec le même filtre à moyenne mobile de période d'intégration $T_c = 1$ s. Une estimation suffit, concentrez-vous sur l'emplacement des maxima et minima s'il y en a.



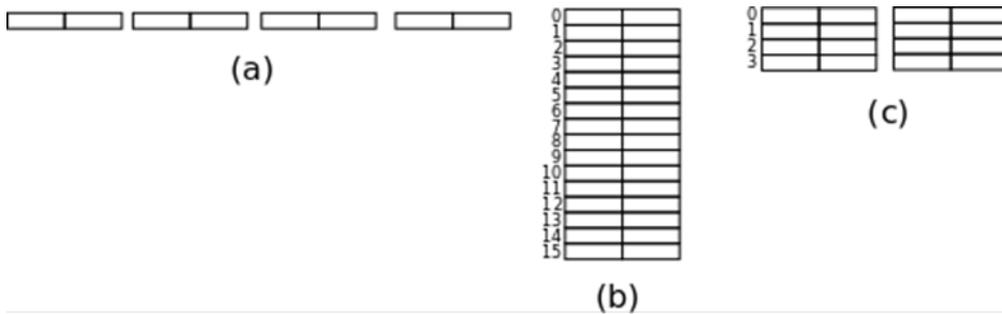
Question Ouverte 2 : (8 pts) Mémoire cache

Dans cet exercice, on cherche à comparer les performances de 3 types d'organisation de la mémoire cache pour différents scenario d'accès à la mémoire centrale. Dans tous les cas examinés ici, **un bloc a toujours une taille de 2 mots**.

Définition : l'adresse d'un bloc en mémoire centrale est notée A_B . Si ce bloc contient le mot d'adresse A_W en mémoire centrale, on pose que : $A_B = A_W/2$ (division entière).

Exemples : si A_W vaut 18 on obtient $A_B = 9$, tandis que pour $A_W = 5$ l'adresse du bloc qui contient ce mot est $A_B = 2$.

Les trois types organisations de la mémoire cache sont illustrés ci-dessous :

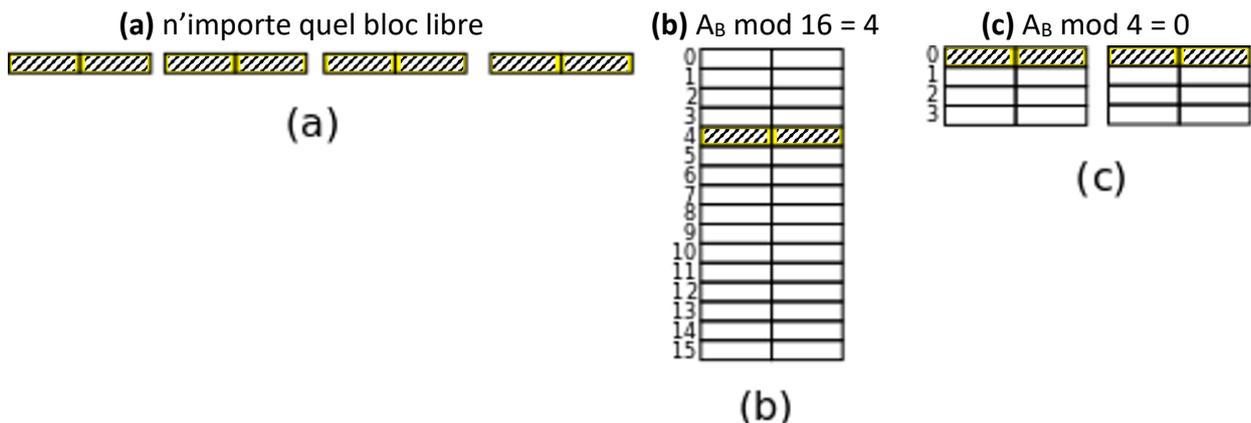


(a): 4 blocs indépendants, sans contrainte sur le rangement d'un bloc en mémoire cache : n'importe quel bloc libre peut être utilisé pour ranger un bloc provenant de la mémoire centrale. Quand le cache est plein, le remplacement est effectué selon l'approche **LRU** (Least Recently Used): c'est le bloc le moins récemment utilisé qui est remplacé.

(b): 16 blocs numérotés de 0 à 15, avec une contrainte sur le rangement d'un bloc en mémoire cache : le bloc de la mémoire centrale d'adresse A_B ne peut être rangé *que* dans le bloc de la mémoire cache dont le numéro est $A_B \bmod 16$.

(c): 8 blocs organisés en deux groupes de 4 blocs numérotés de 0 à 3 : c'est une solution hybride entre les deux précédentes. Ainsi un bloc d'adresse A_B en mémoire centrale peut être rangé à deux emplacements dont le numéro est $A_B \bmod 4$. Quand les deux seuls emplacements autorisés sont occupés, un remplacement de type LRU est effectué mais seulement sur les 2 emplacements possibles.

Supposons qu'on veuille lire le mot d'adresse $A_W = 9$. Le bloc qui contient ce mot a pour adresse de bloc $A_B = 4$. Sachant cela, le bloc sera rangé en mémoire cache dans l'un des blocs hachurés de la mémoire cache :



2.1) Compléter les tables suivantes pour chacun des types de mémoire cache et pour deux séquences indépendantes de demandes de **lecture de mots** de la mémoire centrale. Pour chacune des adresses successives de mots indiquer s'il y a un **échec** (noté **M** comme *Miss*) ou un **succès** (noté **H** comme *Hit*). Les deux premières lignes des tables sont déjà remplies à titre d'exemple pour la première séquence.

Sequence de lecture de **mots 1**) 0, 1, 16, 17, 1, 33, 16, 0, 32

Organization: (a)

A_W	$A_B = A_W/2$	hit (H) or miss (M)?
0	0	M
1	0	H
16		
17		
1		
33		
16		
0		
32		

Total des échecs :

(b)

A_W	$A_B = A_W/2$	numéro = $A_B \bmod 16$	hit (H) or miss (M)?
0	0	0	M
1	0	0	H
16			
17			
1			
33			
16			
0			
32			

.....

(c)

A_W	$A_B = A_W/2$	numéro = $A_B \bmod 4$	hit (H) or miss (M)?
0	0	0	M
1	0	0	H
16			
17			
1			
33			
16			
0			
32			

.....

Sequence de lecture de **mots 2**) 0, 2, 4, 6, 8, 10, 16, 1, 3, 5, 7

Organization: (a)

A_W	$A_B = A_W/2$	hit (H) or miss (M)?
0		
2		
4		
6		
8		
10		
16		
1		
3		
5		
7		

Total des échecs:

(b)

A_W	$A_B = A_W/2$	numéro = $A_B \bmod 16$	hit (H) or miss (M)?
0			
2			
4			
6			
8			
10			
16			
1			
3			
5			
7			

.....

(c)

A_W	$A_B = A_W/2$	numéro = $A_B \bmod 4$	hit (H) or miss (M)?
0			
2			
4			
6			
8			
10			
16			
1			
3			
5			
7			

.....

2.2) Un type d'organisation de la mémoire cache est-il plus performant, indépendamment des séquences de demandes de lectures ?