(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

---

**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 connait 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 connait 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 Huffmann ?

QUIZZ++

| | |
|---|---|
| A | Aucune des autres réponses |
| B | 2 |
| C | 4 |
| D | 8 |

**Complément exigé** : donner les **codes** supplémentaires

1100
1101
1110
1111

---

**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) |

(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

---

**Algo1**

entrée : entier **N** > 2 , Liste L contenant N entiers
sortie : aucune (affiche des valeurs)

```
x ← 0
y ← 0
k ← 0
Pour i de 2 à N-1
    Si L(i) < L(i-1)
        Si L(i) < L(i+1)
            k ← k+1
            Si x = 0
                min_min ← L(i)
                Afficher min_min
                x ← 1
            Sinon
                Tant que (L(i) < min_min)
                    min_min ← L(i)
                    Afficher min_min
            Si y = 0
                max_min ← L(i)
                Afficher max_min
                y ← 1
            Sinon
                Tant que (L(i) > max_min)
                    max_min ← 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 |

L'algorithme détecte les minima locaux d'une suite de valeurs et elle mémorise le minimum des minima ainsi que le maximum des minima.

**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 $O(N^2)$ |

La complexité est linéaire car il y a un seul passage dans les boucles Tant-Que (coût constant de ces boucles).

(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

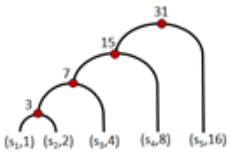
---

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 connait 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 |

Exemple avec 5 caractères différents et les nombres d'apparitions : 1,2,4,8,16.



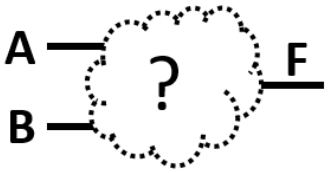

**Question 7 (2 pts)** : Pour **quelle configuration d'apparitions** des N caractères différents obtient-on la plus grande longueur de code possible ?

| | |
|---|---|
| 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 |

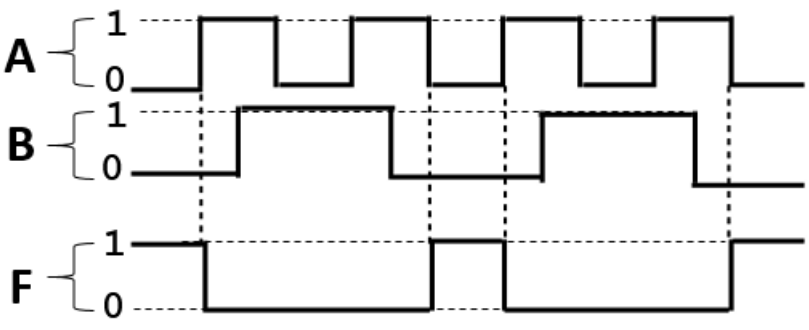
---

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

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.




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.




QUIZZ++

Quelle est la porte logique ?

| | |
|---|---|
| A | Aucune des autres réponses |
| B | NAND |
| C | NOR |
| D | XNOR |

**Complément exigé** : donner la table de vérité

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

---

**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 $O(\log(n))$ |

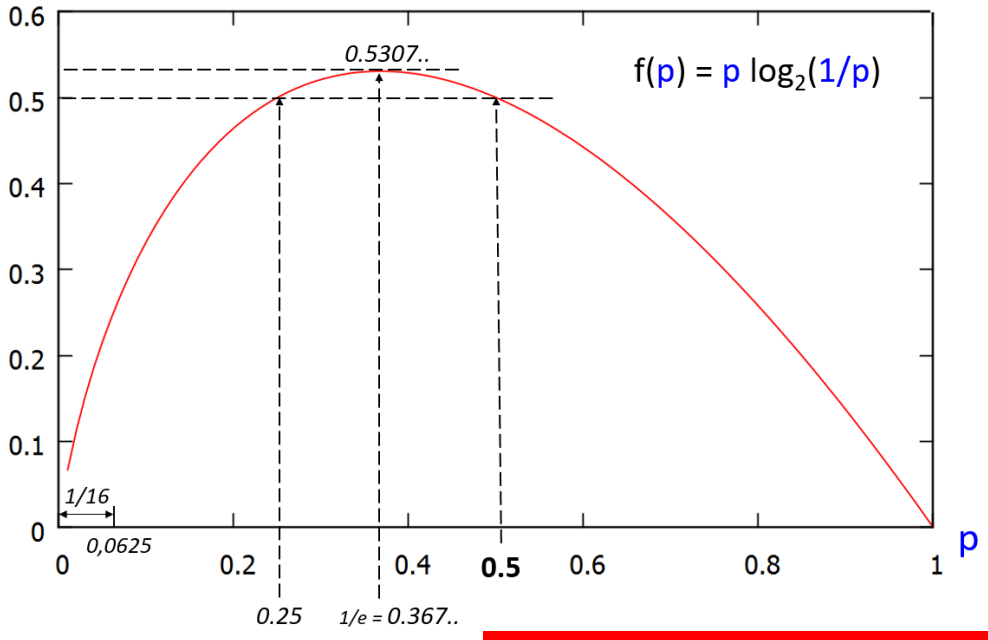
---

**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 :

| N° séqu | séquence |
|---|---|
| **S1** | **AAAARRRRGGGHHHHH** |
| **S2** | **AAAARRRRGGGGHHHH** |
| **S3** | **AAAARRRRGGHHHHHH** |
| **S4** | **AAARRRRRGGHHHHHH** |

Il n'est pas possible de calculer explicitement toutes les entropies car la fonction log2 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 décroit. 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.




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

L'entropie maximum vaut 2

car chacune des 4 lettres est présente avec la même probabilité de 0,25. Chaque terme de l'entropie donne 0,5. Au total on a H = 4 x 0,5

(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

---

**Question 11++ (4 pts)** : Echantillonnage

Soit le signal X(t) = X1(t) + X2(t) + X3(t)

Avec : $X1(t) = -68\sin^2(30\pi t)$ ; $X2(t) = 34\sin(30\pi t)$ ; $X3(t) = 34$

On désire échantillonner X(t) avec une fréquence **fe= 33 Hz.**

QUIZZ++ : quelle proposition est vraie ?

| | |
|---|---|
| A | X(t) a une composante continue nulle et il y a sous-échantillonnage à cause de X1(t) |
| B | X(t) est la fonction nulle |
| C | Il y a sous-échantillonnage à cause de X2(t) |
| D | Il y a sous-échantillonnage à cause de X3(t) |

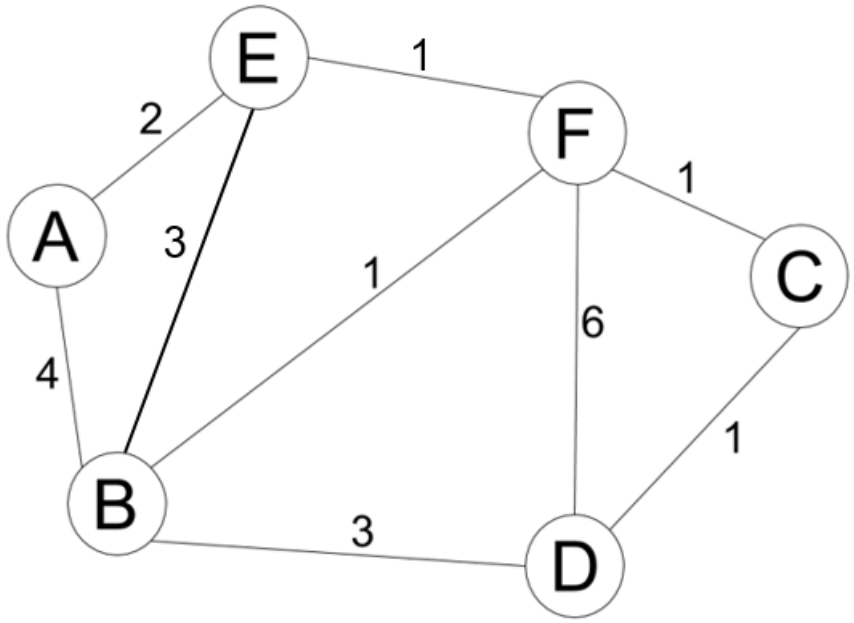
**Complément exigé** : pourquoi ?

- $X1(t) = -68/2(1 - \cos 60\pi t)$
  $= -34 + 34\cos(60\pi t)$
- La valeur -34 annule X3(t)
- la fréquence de $\cos(60\pi t)$ est : 30Hz > fe/2

---

**Question 12 (2 pts)**: Routage

Pour ce réseau, la plus courte « distance » indiquée dans la table est la somme des poids indiqués à coté des liens entre les noeuds du réseau. Par exemple la distance entre A et B est 4.




Indiquer quelle est la **table de routage du nœud A** parmi les choix suivants. Pour chaque **destination** (colonne de gauche) on indique la **direction** du prochain nœud (colonne du milieu) et la **distance** du plus court chemin (colonne droite)

| A | | |
|---|---|---|
| **dest.** | **dir.** | **dist.** |
| B | B | 4 |
| C | E | 4 |
| D | E | 9 |
| E | E | 2 |
| F | E | 5 |

| A | | |
|---|---|---|
| **dest.** | **dir.** | **dist.** |
| B | B | 4 |
| C | E | 5 |
| D | B | 7 |
| E | E | 2 |
| F | B | 5 |

| A | | |
|---|---|---|
| **dest.** | **dir.** | **dist.** |
| B | B | 4 |
| C | E | 4 |
| D | E | 5 |
| E | E | 2 |
| F | E | 3 |

| A | | |
|---|---|---|
| **dest.** | **dir.** | **dist.** |
| B | B | 4 |
| C | B | 6 |
| D | B | 7 |
| E | E | 2 |
| F | B | 5 |

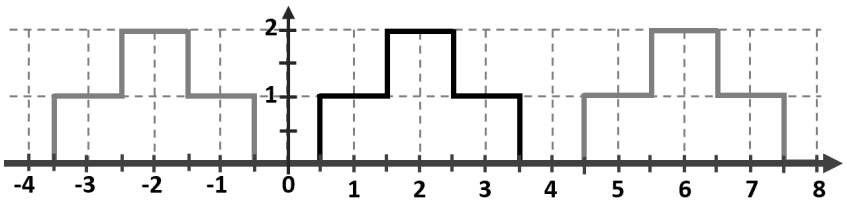
Réponse : A. B. C. D.

(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

---

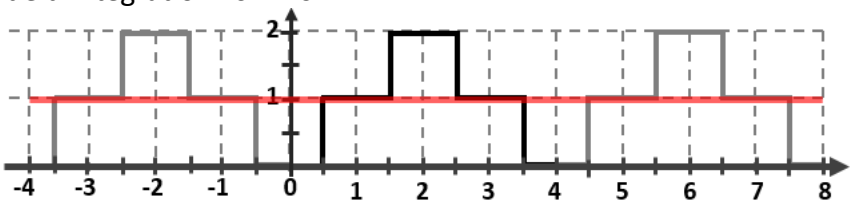
## ***Question Ouverte 1 : (7 pts)*** *Filtre à moyenne mobile*

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



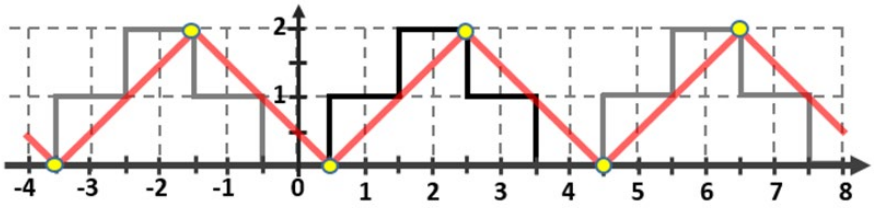

Les 3 questions suivantes demandent d'effectuer un filtrage à moyenne mobile. Pour **chaque** réponse on demande de justifier la forme du signal filtré en indiquant quel ***type d'équation*** permet de le représenter (ex : le signal initial est construit par morceau, chaque morceau est de la forme y= 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 **Tc = 4 s.**



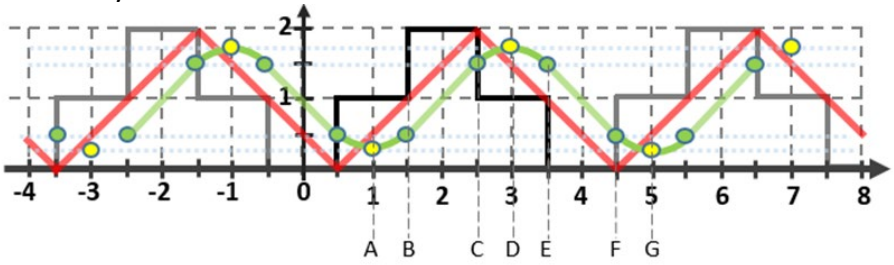

Le signal filtré est la fonction constante y = 1. car **Tc = T**

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 **Tc = 1 s**.




Le signal filtré est de période T et linéaire par morceaux (de la forme y= a.x+b) .
Le minimum vaut zéro ; il est obtenu pour x=0,5. Le maximum vaut 2 ; il est obtenu pour x=2,5.

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 **Tc=1s.** Une estimation suffit, concentrez-vous sur l'emplacement des maxima et minima s'il y en a.




Le signal filtré est de période T. Les segments BC et EF sont linéaires (Y=a.x+b) car la surface augmente (ou diminue) d'une quantité constante pour chaque incrément dx. Cela n'est pas le cas pour les segments AB, CD, DE et FG qui sont des paraboles ($Y= ax^2+bx+c$).
Le minimum est A (1 ; 0,25). Le maximum est D(3 ; 1,75). Les autres points importants sont B(1,5 ; 0,5), C(2,5 ; 1,5), E(3,5 ; 1,5), F(4,5 , 1,5). Chaque filtrage successif conserve la fréquence mais diminue l'amplitude du signal autour de la valeur moyenne 1 et introduit chacun un léger déphasage ; il y a *lissage* du signal.

(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

---

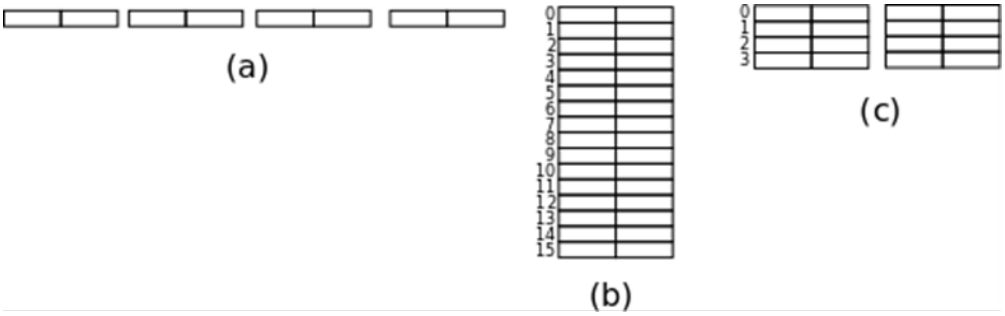
## *Question Ouverte 2 : (8 pts)* *Mémoire cache*

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 $\mathbf{A_B}$ . Si ce bloc contient le mot d'adresse $\mathbf{A_W}$ en mémoire centrale, on pose que : $\mathbf{A_B = A_W/2}$ (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 :

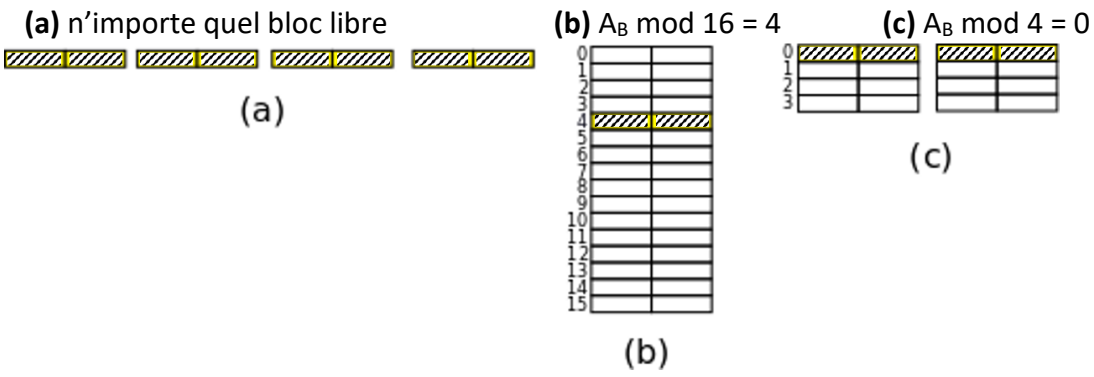



**(a)**: **4 blocs indépendants**, sans contrainte sur le rangement d'un bloc en mémoire cache : n'importe quel bloc libre peut être utilisé pour ranger un bloc provenant de la mémoire centrale. Quand le cache est plein, le remplacement est effectué selon l'approche **LRU** (Least Recently Used): c'est 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 $\mathbf{A_B}$ ne peut être rangé *que* dans le bloc de la mémoire cache dont le numéro est $\mathbf{A_B}$ **mod 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 $\mathbf{A_B}$ en mémoire centrale peut être rangé à deux emplacements dont le numéro est $\mathbf{A_B}$ **mod 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 :



(le texte surligné en vert fournit un complément, non demandé, pour certaines questions)

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 | 8 | M |
| 17 | 8 | H |
| 1 | 0 | H |
| 33 | 16 | M |
| 16 | 8 | H |
| 0 | 0 | H |
| 32 | 16 | H |

Total des échecs : ...3...

(b)

| $A_W$ | $A_B = A_W/2$ | numéro = $A_B$ mod 16 | hit (H) or miss (M)? |
|---|---|---|---|
| 0 | 0 | 0 | M |
| 1 | 0 | 0 | H |
| 16 | 8 | 8 | M |
| 17 | 8 | 8 | H |
| 1 | 0 | 0 | H |
| 33 | 16 | 0 | M |
| 16 | 8 | 8 | H |
| 0 | 0 | 0 | M |
| 32 | 16 | 0 | M |

.....5...

(c)

| $A_W$ | $A_B = A_W/2$ | numéro = $A_B$ mod 4 | hit (H) or miss (M)? |
|---|---|---|---|
| 0 | 0 | 0 | M |
| 1 | 0 | 0 | H |
| 16 | 8 | 0 | M |
| 17 | 8 | 0 | H |
| 1 | 0 | 0 | H |
| 33 | 16 | 0 | M |
| 16 | 8 | 0 | M |
| 0 | 0 | 0 | M |
| 32 | 16 | 0 | M |

...6...

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 | 0 | M |
| 2 | 1 | M |
| 4 | 2 | M |
| 6 | 3 | M |
| 8 | 4 | M |
| 10 | 5 | M |
| 16 | 8 | M |
| 1 | 0 | M |
| 3 | 1 | M |
| 5 | 2 | M |
| 7 | 3 | M |

Total des échecs: ...11...

(b)

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

.....7.....

(c)

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

.....8...

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

Non, la performance dépend de l'application. (a) est la meilleure pour la séquence 1 qui est une illustration de la localité temporelle et la pire pour la localité spatiale (séquence 2).