

Aucun document ou support autre que le sujet ou les copies d'examen n'est autoris .  
 (la copie ou les brouillons du voisin ne sont pas des supports autoris s).  
  teignez imp rativement vos mobiles.

Lorsque des calculs sont n cessaires, il est imp ratif de les pr senter sur la feuille d'examen. Il est aussi n cessaire de **justifier** ses r ponses.

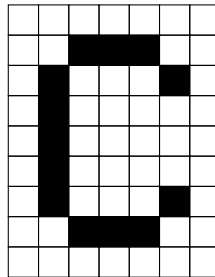
## 1 Exercice (arithm tique)

Soit les mots de 7 bits suivants  $m_1 = 1111110$  et  $m_2 = 0100100$ .

1. S'il s'agit de nombres non sign s sur 7 bits, quelles sont leurs valeurs exprim es en base 10 ? en base 4 ?
2. S'il s'agit de nombres sign s en compl ment   deux sur 7 bits, quelles sont leurs valeurs exprim es en base 10 ? en base 4 ?
3. S'il s'agit de nombres sign s en compl ment   deux sur 7 bits, calculer leur somme en base 2 ?
4. S'il s'agit de nombres non sign s sur 7 bits, calculer leur produit en base 2 (le r sultat doit  tre sur 14 bits).

## 2 Exercice (codage)

Soit la repr sentation, en image noir et blanc, de la lettre C :



On rappelle que le principe du codage RLE (ici appliqu    des suites de 0 et 1) consiste   coder une suite de symboles identiques par leur longueur, par exemple 000000111 sera cod  par 7, 3. Dans ce codage on consid re que l'on commence toujours par des 0, donc le codage du mot 110000 sera cod  0, 2, 4. Dans la suite on supposera que les dimensions de l'image sont connues.

1. Sans codage combien de bits faut-il pour repr senter l'image ? On utilisera 0 pour repr senter la couleur blanche et 1 pour la couleur noire.
2. Utiliser le codage RLE pour coder l'image en codant **ind pendamment chaque ligne** et dans l'ordre de la premi re   la derni re et de gauche   droite. Fournir la suite des valeurs.
3.  crire une fonction Java qui permet   partir de la suite des longueurs RLE donn e sous la forme d'un tableau et de la largeur de l'image de reconstituer l'image avec des 0 et des 1. Le prototype de la fonction sera `void decode(int largeur, int []rle)`
4. Si l'on code les entiers de la suite comme des mots binaires de taille fixe, quelle sera la taille du code ? Quelle sera la taille du codage de la suite (c'est- -dire de l'image) ? Dans notre cas, ce codage permet-il de compresser ?
5.   partir de la suite RLE, donner le tableau associant   chaque valeur  $y$  apparaissant, son nombre d'occurrences (c'est- -dire le nombre de fois que cette valeur est r p t e).
6.   partir du tableau construire l'arbre de Huffman correspondant. Prendre soin de pr ciser les poids de chaque n ud, les valeurs de la suite associ es aux feuilles et les  tiquettes associ es aux arcs.
7.   partir de cet arbre, coder la suite avec le code de Huffman. Quelle est la taille du codage de l'image ? Dans ce cas, le double codage RLE puis Huffman permet-il de compresser ?

### 3 Exercice (codage, cryptographie)

Pour limiter la triche aux examens, vos enseignants ont inventé un exercice crypto-personnalisé. Pour qu'un étudiant puisse fournir une réponse il doit utiliser comme clé de cryptage la première lettre de son nom de famille (ex. : si son nom de famille est «Yunès», il doit utiliser Y (en majuscules)). On rappelle les éléments utiles de la table ASCII :

Code hexa.	Lettre	Code hexa.	Lettre	Code hexa.	Lettre
41	A	42	B	43	C
44	D	45	E	46	F
47	G	48	H	49	I
4A	J	4B	K	4C	L
4D	M	4E	N	4F	O
50	P	51	Q	52	R
53	S	54	T	55	U
56	V	57	W	58	X
59	Y	5A	Z		

Le codage d'un mot  $a_0a_1a_2 \dots a_{n-1}$  écrit avec l'alphabet des 26 lettres consiste alors à appliquer la fonction  $c$  :

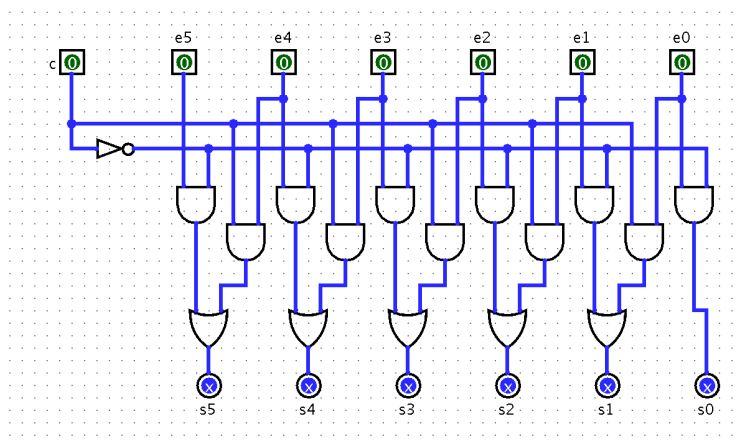
$$\begin{cases} c(a_0) = b(a_0) \oplus b(x) \\ \forall i \in [1, n[, c(a_i) = b(a_i) + c(a_{i-1}) \end{cases}$$

où  $\oplus$  désigne le ou-exclusif bit-à-bit sur les représentations binaires des valeurs,  $x$  désigne la première lettre de votre nom de famille et  $b(y)$  le code ASCII sur 7 bits de la lettre  $y$ .

1. Quelle est la première lettre de votre nom de famille (attention toute valeur qui se révélera «fausse» annulera l'évaluation de l'exercice en entier) ?
2. Quel est son code ASCII écrit en binaire sur 7 bits ?
3. Coder le mot PFUN.

### 4 Exercice (circuit, logique)

Soit le circuit suivant :



1. Si  $e_3 = 1$  et  $c = 0$  que vaut  $s_3$  ?
2. Si  $e_4 = 1$  et  $c = 1$  que vaut  $s_5$  ?
3. Écrire en fonction de  $c, e_i$  ( $i \in [0, 5]$ ) les fonctions correspondantes aux sorties  $s_i$  ( $i \in [0, 5]$ )
4. Quelle est la fonction calculée par le circuit lorsque  $c = 0$  ? et lorsque  $c = 1$  ?
5. Quel est l'opérateur Java correspondant à la fonction lorsque  $c = 1$  (s'il y avait 32 bits en non 6 en entrée) ?

## 5 Exercice (logique, circuit)

On cherche à modéliser le système de contrôle de feux de circulation tricolores. On rappelle que tout feu possède trois lampes, une verte, une orange, une rouge. D'autre part, un feu passe de vert à orange, d'orange à rouge puis de rouge à vert. À un carrefour on a deux paires de feux synchronisés et les feux d'une paire sont toujours de même couleur.

Pour faire varier les feux on utilise une horloge  $t$ , celle-ci fournit au cours du temps et successivement 16 valeurs (les entiers de 0 à 15) en bouclant indéfiniment : 0, 1, ..., 15, 0, 1, ... Ces valeurs sont représentées sur 4 bits  $t_3t_2t_1t_0$ .

Pour la paire de feux  $F_1$ , les feux sont rouges pour toutes les valeurs d'horloge dans l'intervalle  $[0, 7]$  (on notera ce prédicat  $R_1$ ), verts dans l'intervalle  $[8, 13]$  (prédicat  $V_1$ ) et oranges dans l'intervalle  $[14, 15]$  (prédicat  $O_1$ ).

Pour la paire de feux  $F_2$ , les feux sont verts dans l'intervalle  $[0, 6]$  (prédicat  $V_2$ ), oranges dans l'intervalle  $[7, 8]$  (prédicat  $O_2$ ) et rouges dans l'intervalle  $[8, 15]$  (prédicat  $R_2$ ).

1. Donner la table de vérité regroupant les prédicats  $V_1, O_1, R_1, V_2, O_2$  et  $R_2$  en fonction de l'horloge exprimée sur 4 bits. C'est-à-dire sous la forme :

$t_3$	$t_2$	$t_1$	$t_0$	$V_1$	$O_1$	$R_1$	$V_2$	$O_2$	$R_2$
0	0	0	0						
⋮									
1	1	1	1						

2. En lisant simplement le tableau précédent, donner une écriture simplifiée des fonctions  $R_1(t)$  et  $R_2(t)$ .
3. En utilisant la méthode de Karnaugh simplifier les fonctions  $V_1(t)$  et  $V_2(t)$ .
4. Simplifier par la méthode que vous voulez (une simple lecture de la table devrait suffire) les fonctions  $O_1(t)$  et  $O_2(t)$ .
5. Dessiner le circuit qui prend en entrée les 4 bits de l'horloge et fournit en sortie les fonctions des différentes lampes des feux.