Partiel	PF1	24/10/2018 10^{50}–12^{40}

aucun document – aucune machine

quatre exercices indépendants – barème indicatif

La qualité de la rédaction et la clarté de la présentation seront prises en compte dans la notation : les réponses devront être précises et argumentées.

Exercice 1 (4 points)

Pour toutes les bases plus grandes que 10, on utilisera comme chiffres complémentaires les lettres de l'alphabet, soit `0123456789ABCDEFG`...

1. $(\texttt{84AB45D34561A2})_{14}$ est-il divisible par 2 ? par 7 ? $(\texttt{123456789ABC7})_{14}$ est-il divisible par 2 ? par 7 ?
2. Convertir en base 2 le nombre $(\texttt{972})_{10}$.
3. Calculer en base 2 la somme de `110111` et de `11001001`, poser le calcul avec les retenues.
4. Calculer en base 2 l'expression $\texttt{1011011} - (\texttt{1101} \times \texttt{110})$, poser le calcul avec les retenues.

Exercice 2 (4 points)

Dans cet exercice, on s'intéresse à la représentation des nombres en base 2 sur 8 bits.

1. Si la représentation est signée en complément à 2, quels sont les nombres représentés ? Donner l'intervalle.
2. Dans la représentation signée en complément à 2, quels sont les codages des nombres $(\texttt{56})_{10}$ et $(-\texttt{74})_{10}$?
3. Dans la représentation signée en complément à 2, à quel nombre en base 10 correspond le codage `11001000` ?
4. Effectuer dans la représentation signée en complément à 2 les trois opérations suivantes (en précisant à chaque fois si le résultat peut être considéré comme correct dans l'arithmétique ordinaire) :

$$\texttt{00010100} - \texttt{01101111},$$

$$\texttt{11110100} + \texttt{11101000},$$

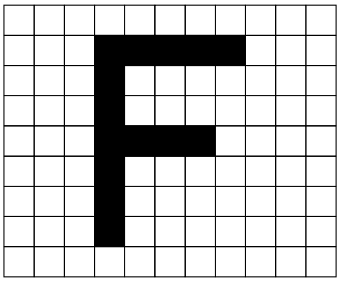
$$\texttt{11111111} \times \texttt{11111001}.$$

Exercice 3 (4 points)

1. Pour quelle valeur de r l'expression suivante est-elle vraie ? $(\texttt{121})_r = (\texttt{144})_8$.
2. Pour quelle valeur de r l'expression suivante est-elle vraie ? $(\texttt{33})_r + (\texttt{21})_r = (\texttt{24})_{10}$.
3. Pour quelle valeur de r l'expression suivante est-elle vraie ? $(\texttt{23})_r + (\texttt{25})_r = (\texttt{51})_r$.
4. Pour quelle valeur de r l'expression suivante est-elle vraie ? $(\texttt{423})_r - (\texttt{154})_r = (\texttt{236})_r$.

Exercice 4 (8 points)

Soit la représentation, en image noir et blanc, de la lettre F :



On rappelle que le principe du codage RLE (ici appliqué à des suites de `0` et `1`) consiste à coder une suite de symboles identiques par leur longueur, par exemple `0000000111` est codé par 7, 3. Dans ce codage, on considère que l'on commence toujours par des `0`, donc le codage du mot `110000` est 0, 2, 4. Dans la suite, on suppose que les dimensions de l'image sont connues.

1. Sans codage combien de bits faut-il pour représenter l'image ? On utilise `0` pour représenter la couleur blanche et `1` pour la couleur noire.
2. Utiliser le codage RLE pour coder l'image en codant **indépendamment chaque ligne** de gauche à droite, et dans l'ordre de la première à la dernière ligne. Fournir la suite des valeurs.
3. Si l'on code les entiers de la suite comme des mots binaires de taille fixe, quelle est la taille d'un mot de code ? Quelle est la taille du codage de la suite (c'est-à-dire de l'image) ? Dans notre cas, ce codage permet-il de compresser ? Si oui, quel est le taux de compression obtenu ?
4. À partir de la suite RLE, donner le tableau associant à chaque valeur y apparaissant, son nombre d'occurrences (c'est-à-dire le nombre de fois que cette valeur est répétée).
5. À partir du tableau, construire l'arbre de Huffman correspondant. Prendre soin de préciser le poids de chaque nœud, la valeur associée à chaque feuille et l'étiquette associée à chaque arc.
6. À partir de cet arbre, coder la suite RLE avec le code de Huffman. Quelle est la taille du codage de l'image ? Dans ce cas, le double codage (c'est-à-dire, RLE puis Huffman) permet-il de compresser ? Si oui, quel est le taux de compression obtenu ?