Corrigé et barème – Données numériques : bit, binaire et codage

Le bit et les booléens ; lire et écrire un nombre entier en binaire ; coder et décoder du texte avec le code ASCII (ASCII étendu, UTF-8) ; suivre et lire les données mesurées puis téléversées par un objet

Durée : 55 min – Sans document – Sur 20 points · Technologie 4e

Exercice 1 – Le parking souterrain : des booléens dans un octet ★★ *(4 pts)*

a) Cinq bits valent 1, trois valent 0 : **3 places libres**, les places 2, 5 et 8. (1)

b) Places 1, 2 et 8 à 1, les autres à 0 : **11000001**. (1)

c) Poids des 1 : $128 + 32 + 16 + 4 + 2 = \mathbf{182}$. (1)

d) 20 places → **20 bits** (un par place) ; 2 octets n'en contiennent que 16 : il faut **3 octets** (24 bits, dont 4 inutilisés). Combinaisons : $2^{20} = 2^{10} \times 2^{10} = 1\,024 \times 1\,024 = \mathbf{1\,048\,576}$. (1)

Erreur fréquente — Le nombre de bits (un par place) n'est pas le nombre de combinaisons (2^n) : 20 places demandent 20 bits, pas un million.

Exercice 2 – Conversions entre décimal et binaire ★★ *(4 pts)*

a) $(1011011)_2 = 64 + 16 + 8 + 2 + 1 = \mathbf{91}$; $(11110001)_2 = 128 + 64 + 32 + 16 + 1 = \mathbf{241}$. (1)

b) $140 = 128 + 8 + 4 \rightarrow \mathbf{10001100}$; $29 = 16 + 8 + 4 + 1 \rightarrow \mathbf{00011101}$. (1)

c) $2^9 = 512$ ne dépasse pas 700, $2^{10} = 1\,024$ le dépasse : **10 bits** (sur 9 bits, on s'arrête à 511). (1)

d) $42 \div 2 : 21$, il reste 0 ; $21 \div 2 : 10$, il reste 1 ; $10 \div 2 : 5$, il reste 0 ; $5 \div 2 : 2$, il reste 1 ; $2 \div 2 : 1$, il reste 0 ; $1 \div 2 : 0$, il reste 1. Lucas a lu les restes dans l'ordre où il les a écrits, ce qui donne $(010101)_2 = 21$. Lus du dernier au premier : $42 = \mathbf{(101010)_2}$. Vérification : $32 + 8 + 2 = 42$. (1)

Le point clé — Une conversion se vérifie toujours en refaisant le chemin inverse : ici, l'erreur de Lucas donne 21, la moitié de 42, et l'addition des poids l'aurait montré tout de suite.

Exercice 3 – Un message codé en ASCII ★★★ *(4 pts)*

a) $(01011001)_2 = 64 + 16 + 8 + 1 = 89 = 64 + 25$: Y ; $(01100101)_2 = 64 + 32 + 4 + 1 = 101 = 96 + 5$: e ; $(01110011)_2 = 64 + 32 + 16 + 2 + 1 = 115 = 96 + 19$: s. Message : « **Yes** ». (1)

b) F $= 64 + 6 = 70 \rightarrow 01000110$; a $= 97 \rightarrow 01100001$; x $= 96 + 24 = 120 \rightarrow 01111000$. « Fax » = **01000110 01100001 01111000**. (1)

c) « Bonne fête » compte 10 caractères, l'espace compris : **10 octets** en Latin-1 ; en UTF-8, le « ê » prend 2 octets : **11 octets**. L'ASCII étendu met les lettres accentuées sur un seul octet, UTF-8 sur deux. (1)

d) H $= 64 + 8 = 72 \rightarrow 01001000$; h $= 96 + 8 = 104 \rightarrow 01101000$. Seul le 3e bit en partant de la gauche change : c'est le bit de poids **32**, car minuscule = majuscule + 32. (1)

Attention — L'espace est un caractère comme un autre : il compte dans la taille d'un texte.

Exercice 4 – L'aquarium connecté : lire une table téléversée ★★★ *(4 pts)*

a) **7 enregistrements**, **3 descripteurs** : heure (chaîne de caractères), eau (nombre), chauffage (booléen). 7 relevés → 6 intervalles de 20 min : **120 min**, soit 2 h. (1)

b) $25 + 24 + 23 + 25 + 26 + 27 + 25 = 175$; $175 \div 7 = \mathbf{25}$ **°C**. Écart : $27 - 23 = \mathbf{4}$ **°C**. (1)

c) Chauffage = 1 pour 3 relevés : $3 \times 20 = \mathbf{60}$ **min**. D'après la table, il s'allume quand l'eau est descendue à 23 °C et reste en marche pendant que l'eau remonte (25 °C puis 26 °C) ; il est arrêté au relevé où elle atteint 27 °C. (1)

d) « 18:40;23;1 » = $5 + 1 + 2 + 1 + 1 = 10$ caractères, plus la fin de ligne : 11 octets ; $7 \times 11 = \mathbf{77}$ **octets**. En binaire : $8 + 1 = 9$ bits par relevé, $7 \times 9 = \mathbf{63}$ **bits** (moins de 8 octets). (1)

Repère — n relevés réguliers couvrent n − 1 intervalles ; et un nombre écrit en texte prend un octet par caractère.

Exercice 5 – Vrai ou faux argumenté ★★★ *(4 pts)*

a) Faux : sur 4 bits, le plus grand nombre est $(1111)_2 = 8 + 4 + 2 + 1 = 15$; 16 demande un 5e bit. (1)

b) Faux : le texte « 255 » compte 3 caractères, donc 3 octets ; le nombre $255 = 11111111$ tient dans 1 octet. (1)

c) Faux : un bit par lampe suffit, donc 16 bits (2 octets) ; $2^{16} = 65\,536$ est le nombre de combinaisons possibles. (1)

d) Vrai : $(1000000)_2 = 64$ et $(100000)_2 = 32$, et $64 = 2 \times 32$; ajouter un 0 à droite double le nombre. (1)

Astuce — Pour juger une affirmation sur le binaire, on teste la valeur limite : le plus grand nombre sur n bits est $2^n - 1$.

Ce document est la propriété intellectuelle de Ben, professeur particulier (benprofparticulier.fr), protégée par le Code de la propriété intellectuelle (art. L. 111-1 et L. 122-4). Il est mis à disposition pour un usage personnel de révision. Toute reproduction, diffusion, mise en ligne ou revente, totale ou partielle, sans autorisation écrite est interdite et constitue une contrefaçon (art. L. 335-2).