

Principes de fonctionnement des machines binaires

Durée : 3h. Les documents et les appareils électroniques de toutes sortes ne sont pas autorisés.

Toutes vos réponses doivent être justifiées. Les exercices sont indépendants, et peuvent être traités dans un ordre quelconque. Il est demandé de traiter les exercices 1 à 7, et deux exercices au choix parmi les exercices 8 à 11.

Partie obligatoire

Exercice 1 :

Soit le nombre n dont la représentation en base huit est $n = (72540)_8$.

Question 1 : Quelle est la plus grande puissance p_1 de 8 qui divise n ?

Question 2 : Quelle est la plus grande puissance p_2 de 2 qui divise n ?

Question 3 : Quelle est la plus grande puissance p_3 de 16 qui divise n ?

Question 4 : Quels sont le quotient q et le reste r de la division de n par 16^2 ?
Donner les résultats en base seize et en base dix.

Exercice 2 :

Question 1 : Calculer la représentation binaire des nombres 16384, 375 et -16384 , 45, en virgule fixe puis en virgule flottante.

Question 2 : Donner la représentation en machine de ces deux nombres dans le type `float` sous forme d'un nombre hexadécimal.

Exercice 3 :

On veut coder un alphabet coloré composé de 336 caractères (lettres, chiffres, symboles, etc) qui peuvent prendre chacun 6 couleurs différentes, en utilisant un code de longueur constante, sur k bits.

Question 1 : Quelle doit être la valeur minimale de k ?

Question 2 : On suppose maintenant que le code est obtenu en juxtaposant un code de longueur constante k_1 pour le caractère et un code de longueur k_2 pour la couleur. Quelle doit être maintenant la valeur minimale de k ?

Exercice 4 : (Le détective)

L'espace mémoire suivant contient, dans le désordre :

- un `short` négatif suivi de son opposé
- un `float` compris entre 1 et 2
- un `int` positif
- un mot de trois lettres codé en ASCII suivi du caractère terminateur

```
01100010 01100001 01100011 00000000
00000000 00000000 00000001 00101100
00111111 10100100 00000000 00000000
11111100 00011000 00000011 11101000
```

Dire lequel est lequel, en justifiant votre réponse, et retrouver le mot et les valeurs décimales des nombres.

Exercice 5 : (Une chaîne flottante)

Un espace mémoire contient la chaîne de caractères ASCII :

```
"-104.625"
```

Question 1 : Interpréter cet espace mémoire comme une suite de chiffres hexadécimaux.

Question 2 : Cette chaîne contient la représentation décimale d'un nombre rationnel. Vérifier que celui-ci peut être représenté dans le type `float` sans perte de précision et comparer la taille mémoire de ces deux codages de la même donnée.

Rappels Le standard ASCII code les caractères alphanumériques de 0 à 127. Les caractères sont alignés sur 8 bits. On y trouve notamment :

- 0 : le caractère terminateur de chaînes de caractères
- 45 : le signe moins -
- 46 : le point .
- de 48 à 57 : les chiffres de 0 à 9
- de 65 à 90 : l'alphabet des majuscules (A à Z)
- de 97 à 122 : l'alphabet des minuscules (a à z).

Exercice 6 : (Tautologies à peu de variables)

Lorsqu'une expression booléenne est composée de peu de variables, on peut profiter des opérations bits-à-bits de Java pour calculer simultanément les différentes lignes de sa table de vérité.

Question 1 : On affecte les nombres binaires suivants aux variables a , b et c :

```
00001111
00110011
01010101
```

Quel est le plus petit type numérique Java permettant de coder ces nombres ? Donner leurs représentations hexadécimales et décimales.

Question 2 : Calculer le résultat de l'évaluation de l'expression Java $(a\&b) | c$. En considérant ce résultat bit par bit, expliquer comment cela permet de déduire la table de vérité de l'expression booléenne $(a \wedge b) \vee c$.

Question 3 : Calculer de même l'expression Java $(a\&b\&c) | \sim a | \sim b | \sim c$ et donner l'expression booléenne ainsi simulée. Comment peut-on alors vérifier simplement si cette expression correspond bien à une tautologie ? Même question pour une antilogie (c'est-à-dire une expression booléenne toujours fausse).

Question 4 : Expliquer comment utiliser cette méthode pour tester des expressions booléennes contenant des "ou exclusifs", des "implications", des "équivalents" (si et seulement si).

Question 5 : On souhaite maintenant simuler des expressions booléennes à 4 variables. Donner le type Java à utiliser, ainsi que les valeurs à mettre dans a , b , c et d .

Question 6 : Plus généralement, jusqu'à combien de variables booléennes pourra-t-on aller si les entiers de la machine sont au plus sur 32 bits ? sur 64 bits ?

Exercice 7 :

On considère l'expression logique suivante : $(a \Rightarrow (b \oplus c)) \oplus ((c \Rightarrow b) \oplus a)$. (On rappelle que \oplus est le connecteur du "ou" exclusif. \Rightarrow est aussi noté \supset)

Question 1 : Construire l'arbre α associé à cette expression.

Chaque nœud de l'arbre est la racine d'un sous-arbre qui correspond à une sous-expression de l'expression de départ.

Question 2 : Evaluer en chaque nœud de l'arbre la valeur de la sous-expression lui correspondant, pour $a = 0$, $b = 1$ et $c = 1$.

Question 3 : Ecrire le mot w obtenu par le parcours gauche préfixe de cet arbre et le mot f obtenu par le parcours gauche postfixe de cet arbre.

Question 4 : Dresser la table de vérité de l'expression.

Question 5 : En donner la forme normale disjonctive.

Question 6 : Donner sa forme normale conjonctive.

Partie avec choix

Exercice 8 :

Une image, scannée à la résolution de 600 dpi en couleurs vraies pèse 36 Mo (16 millions de couleurs). On suppose qu'il n'y a pas de compression et que ce poids est celui de l'image bitmap obtenue. Or on ne dispose que de 4 Mo pour la stocker.

Question 1 : Est-il possible de la stocker si on la convertit en 256 niveaux de gris ? En 16 niveaux de gris ? En noir et blanc ?

Question 2 : A quelle résolution maximale aurait-on pu la scanner pour que l'on puisse la stocker en couleurs vraies sur 4 Mo ?

Exercice 9 :

On veut tester en utilisant les opérations booléennes bit à bit de Java si l'octet le plus à gauche d'une variable de type `int` contient (au moins) sept "1" qui sont consécutifs, et son octet le plus à droite contient (au moins) sept "0" qui sont consécutifs.

Question 1 : On demande d'écrire en français la suite des opérations à faire pour réaliser ce test.

Question 2 : En déduire une expression utilisant ces opérations booléennes bit à bit de Java qui vaut 1 si ce test est positif, et vaut 0 sinon.

Exercice 10 :

On note \oplus le ou exclusif (`xor`, en java `^`). On rappelle que cette opération est associative. Pour x un entier d'au plus 4 bits, $x = (a_3a_2a_1a_0)_2$, on note $p(x)$ la parité du nombre de bits à 1 de x , soit :

$$p(x) = a_3 \oplus a_2 \oplus a_1 \oplus a_0$$

Question 1 : Soit m le plus grand entier représentable sur 4 bits. Calculer $p(x)$ pour un entier x variant de 0 à m . Présenter les résultats sous forme d'un tableau à double entrée.

Question 2 : Calculer en hexadécimal la constante C dont la représentation binaire est la suite des $p(x)$, x variant de m à 0.

Question 3 : Soit un entier x entre 0 et m (compris). Donner une expression java utilisant C et *au plus* deux opérations bit à bit pour calculer $p(x)$.

Exercice 11 :

Réaliser un circuit prenant en entrée les 4 bits $a_3a_2a_1a_0$ d'un entier n compris entre 0 et 15, et un autre bit b , et qui donne en sortie les 4 bits $s_3s_2s_1s_0$ d'un entier p qui vaut $n/2$ si $b = 0$ et n est pair, $8 + n/2$ si $b = 0$ et n est impair, $2 * n$ si $b = 1$ et $n < 8$, et $2 * n + 1$ si $b = 1$ et $n > 7$.