Université Paris 7- Denis Diderot
L1 d'Informatique — Examen partiel du 24 novembre 2012

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.
Sauf indication contraire, tous les nombres sont donnés en base dix.
On trouvera en fin d'énoncé des rappels qui pourraient être utiles.

Exercice 1 :

Effectuez les changements de base suivants :

Base 2	Base 8	Base 10	Base 16
10011101_2			
	1754_8		
		2913_{10}	
			$A7C_{16}$

On attend le calcul, présenté de manière claire, des valeurs permettant de remplir ce tableau.

Exercice 2 :

Question 1 : Ecrire une table d'addition et une table de multiplication en base 7. En s'aidant de ces tables, effectuer la multiplication suivante : 6543×4015.

Question 2 : En base 16, effectuer l'addition suivante : $B9624AF + 6698CB03$.

Exercice 3 :

Soit r le nombre rationnel dont la représentation en base 10 est $(49,5(6)^\omega)_{10}$.

Question 1 : Donner sa représentation en base 2.

Question 2 : Donner sa représentation en machine en format `float`.

Exercice 4 :

Question 1 : Donner la représentation du nombre -173 en `float`.

Question 2 : Expliquer pourquoi seul l'exposant est modifié par la division de ce `float` par 128 pour la représentation du résultat de cette division, et en déduire cette représentation du résultat de la division de ce `float` par 128.

Question 3 : On définit le codage suivant pour les fractions de la forme $\frac{p}{q}$. Soit `pp` la représentation en `short` de p et `qq` celle de q. On code alors la fraction $\frac{p}{q}$ par la concaténation des représentations en short de p puis q, c'est-à-dire `ppqq`. Donner la représentation de la fraction $\frac{-173}{128}$ avec ce codage.

Question 4 : Donner un exemple de fraction qui soit représentable dans ce codage mais qui *ne* soit *pas* représentable sans perte de précision dans le type `float`.

Question 5 : Donner un exemple de fraction irréductible qui *ne* soit *pas* représentable dans ce codage mais qui soit représentable sans perte de précision dans le type `float`.

Exercice 5 :

Question 1 : Soit τ un code binaire de longueur fixe codant les lettres minuscules et majuscules de notre alphabet (donc codant 52 caractères en tout). Quelle longueur au minimum le code doit-il faire ?

Question 2 : Ecrire en code ASCII le texte suivant : *Bond007*

Exercice 6 :

On scanne une image ayant 24 cm de large et 13,5 cm de hauteur à 300 dpi en noir et blanc.

Question 1 : Poser l'opération qui permet de calculer le poids de cette image. Faire le calcul, sachant qu'un pouce $\simeq 2,54$ cm (on pourra se contenter d'une valeur approchée).

Question 2 : On désire l'afficher sur un écran 27 pouces de format 16/9 ayant une définition de 2560×1440 pixels. La résolution choisie pour scanner l'image est-elle suffisante ?

Rappels :
On rappelle que les types `byte`, `short` et `int` codent les entiers sur 8, 16 et 32 bits respectivement, en utilisant la convention du complément à 2 pour la représentation des nombres négatifs.
On rappelle la représentation en machine du type `float` : 1 bit pour le signe, 8 bits pour l'exposant (valeur de l'exposant vrai augmentée de 127) et 23 bits pour la mantisse.
Le standard ASCII code les caractères alphanumériques de 0 à 127. Les caractères sont alignés sur 8 bits. On y trouve notamment :

- 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).