

## Codage de l'information

### Devoir surveillé n° 1

Vendredi 4 novembre 2016 - Durée 2h - Polycopié de cours autorisé. Calculatrices non autorisées.

Veillez indiquer le numéro de votre groupe de TD sur la copie qu'il est inutile de rendre anonyme, ainsi que votre NIP (figurant sur votre carte d'étudiant).

Ce sujet contient trois exercices indépendants. Prenez 10mn pour lire l'intégralité du sujet avant de commencer.

Évidemment, toute réponse doit être justifiée.

#### Exercice 1-1 *Conversions*

Soit un nombre  $N$  qui s'écrit en hexadécimal  $\overline{Dx7y}_{16}$ , où  $x$  est le chiffre des dizaines de votre NIP (votre NIP en décimal, bien-sûr, ne le convertissez pas) et  $y$  le chiffre des unités.

**Question 1** Donnez la représentation de  $N$  en hexadécimal.

**Question 2** Donnez la représentation de  $N$  en binaire.

**Question 3** En supposant que les 16 bits du nombre  $N$  représentent deux caractères codés en ISO-8859-1, donnez ces deux caractères (voir page 90 du polycopié).

**Question 4** En supposant que ces 16 bits représentent un nombre en complément à 2, donnez la valeur de ce nombre.

**Question 5** Supposons que  $\overline{Dx7y}_{16}$  représente l'état de la mémoire et que la valeur soit stocké en petit boutiste. Quelle est alors la valeur de l'entier  $N$  s'il est représenté en complément à 2 ?

**Question 6** Supposons maintenant que les 16 bits du nombres  $N$ , donnés à la question 2, représentent un nombre flottant. Quelle est la valeur de ce nombre (vous pouvez l'écrire sous forme fractionnaire) ?

#### Exercice 1-2 *Opération logiques*

Dans cet exercice, nous noterons  $r_g(n)$  le rang du bit à 1 de poids le plus fort dans la représentation binaire de  $n$ . La numérotation du rang commence à 0 par le bit de poids le plus faible. Par exemple  $r_g(16) = r_g(18) = 4$ . Dans le cas où  $n = 0$ , on définit que  $r_g(n)$  n'est pas défini.

**Question 1** Donner la valeur de  $r_g(700)$ .

**Question 2** Donnez un algorithme permettant de calculer  $r_g(n)$  pour tout nombre  $n \geq 0$  représenté sur 32 bits, en utilisant exclusivement des opérations logiques (sauf incrémentation ou décrémentation de variables de boucles ou opérateurs de comparaison). L'algorithme renverra `None` dans le cas où  $r_g(n)$  n'est pas défini. Quel est, au maximum le nombre d'itérations effectué par votre algorithme ?

Nous disposons maintenant d'une table  $R$  de 256 entrées telle que  $R[i] = r_g(i)$ .

**Question 3** Ré-écrivez votre algorithme en utilisant la table  $R$  pour réduire le nombre d'étapes, et en ayant toujours recours aux opérations logiques.

**Question 4** À partir du résultat de  $r_g(n)$ , comment calculer  $\lfloor \log_2 n \rfloor$  et  $\lceil \log_2 n \rceil$  ?

### Exercice 1-3 *Être ou ne pas être un code*

#### Question 1

Pour chacun des langages suivants, dites si ce sont des codes. Dans les cas où ce ne sont pas des codes, modifiez un seul bit d'un seul mot du langage, afin que le langage devienne un code. Une modification d'un seul bit revient à inverser sa valeur, les suppressions ou insertions ne sont pas autorisées.

1. {0, 01, 100}
2. {0, 10, 1011, 11010}
3. {01, 011, 0111}
4. {01, 000, 101, 1001}
5. {01, 100, 110, 1100}
6. {101, 000, 111, 010}
7. {10, 100, 0110, 11010}
8. {110, 011, 00101, 10100, 11101}

**Question 2** On souhaite représenter les 128 caractères du codage ASCII avec un code à longueur variable. On souhaite coder 20 caractères avec des mots de longueur 6 ou inférieures et les 108 mots restants avec des mots de longueur  $\ell$  quelconque.

Est-il possible d'avoir un code tel qu'un mot soit de longueur 2, deux mots soient de longueur 3, quatre mots de longueur 4 et treize mots de longueur 6? Aurait-on la possibilité de coder 108 symboles supplémentaires avec une longueur  $\ell$  quelconque? Quelle serait la valeur minimale de  $\ell$ ?