Bit it, bit it, ...

3 février 2016

1 le background

La représentation d'un entier sous forme d'une suite de bits est utile dès lors que l'on doit travailler non pas au niveau de l'octet mais au niveau du bit. Par exemple, les masques de réseau sont définis au niveau du bit. En traitement du signal, on détermine pour chaque échantillon quel est le bit de poids fort et ainsi connaître la taille minimale pour sotcker chaque échantillon. On peut aussi en déduire quel composant électronique choisir pour convertir l'échantillon de l'analogique vers le numérique et inversement.

2 l'énoncé

L'objectif est de réaliser une fonction qui calcule le bits de poids fort **d'un unsigned** int (donc sur 32 bits) et une autre qui calcule le nombre de bits à 1. Pour cela, votre programme doit lire sur l'entrée standard :

- 1. une ligne contenant un entier N, donnant le nombre de valeurs à traiter.
- 2. N lignes, contenant une valeur positive entière non signée (donc entre 1 et $2^{32}-1$).

Pour chaque valeur, votre programme doit :

- calculer l'indice (en partant de 0) du bits de poids fort, c'est-à-dire l'indice du bit à 1 le plus à gauche dans une représentation des bits de droite à gauche.
- calculer le nombre de bits à 1.
- afficher ces deux valuers séparées par un espace sur la sortie standard.

entrée	sortie
3	
99	6 4
123456	16 6
987654321	29 17

Table 1 – Exemple d'entrée et la sortie attendue

Attention : cet exercice se base sur la notation habituelle d'une valeur binaire, en écrivant les bits de droite à gauche. Par exemple, la valeur binaire 100101 correspond à

 $1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 37$. On a donc bien écrit le bit d'indice 0 tout à droite et le bit d'indice 5 tout à gauche.

3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur

http://cours-info.iut-bm.univ-fcomte.fr

un article dans la section hackaton \rightarrow édition 2016, portant le même titre que l'exercice. Il contient un lien permettant de télécharger un canevas de code, ainsi que le fichier d'entrée donné ci-dessus.

Bien entendu, vous êtes libres d'utiliser ou non ce canevas, mais c'est un gain de temps que de s'en servir comme base.