UNIVERSITÉ de Marne-la-Vallée

Travaux Dirigés d'Algorithmique n°3

DEUG MIAS / S4 option info

—Option Informatique—

Écriture bits-à-bits

Dans les algorithmes de compressions de données, on a souvent besoin de stocker des nombres dans un fichier. Nous allons implémenter deux méthodes pour le faire, le but étant de prendre le moins de place possible sur le disque.

La première méthode utilise un nombre variable d'octets, la deuxième utilise un nombre variable de bits. Pour chacune des deux méthodes, on écrira les six fonctions suivantes :

`InitFich`	Ouvre le fichier et initialise les tampons éventuels
`CloseFich`	Ferme le fichier en vidant les tampons éventuels
`EcritEntier`	Écrit un entier dans le fichier
`LitEntier`	Lit un entier dans le fichier
`EcritChar`	Écrit un caractère en clair dans le fichier
`LitChar`	Lit un caractère en clair dans le fichier

▸ Exercice 1. *Écriture sur un nombre variable d'octets*

Dans une première méthode, on découpe l'écriture binaire par paquets de 7 bits, le premier (bit de poids fort) est utilisé pour indiquer si l'écriture se poursuit sur l'octet suivant. Par exemple

- $5 = (101)_2$ *s'écrit sur un octet* **0**0000101 ;
- $707 = (1011000011)_2$ *s'écrit sur deux octets* **1**0000101 **0**1000011 .

▸ Exercice 2. *Écriture sur un nombre variable de bits*

On peut écrire chaque entier n sur un nombre variable de bits, si on indique d'abord le nombre de bits significatifs de l'écriture de n. Cela peut se faire en unaire, c'est à dire par une suite de 1 de longueur égale au nombre de bits utilisés, suivi d'un 0, utilisé comme séparateur.

Le nombre $5 = (101)_2$ est donc transcrit par `111 0 101` *(où* `111` *au début indique que trois bits sont nécessaires, suivi de* `0` *comme séparateur suivi de* `101`*) et 128 est transcrit par* `11111111010000000`.

En fait on peut même faire mieux : les deux séries de bits commencent toujours par un 1 sauf si l'on écrit 0. Pour éviter ce cas à part, on lieu d'écrire n on écrira $n+1$ sans écrire les deux bits 1 sous entendu. Par exemple, pour coder 4, on écrit $5 = (101)_2$ qu'il faudrait coder **1**11 0 **1**01 *et que l'on code donc* 11001.

1. *Pour s'entraîner décodez la suite de bits suivantes :* 11010111010111001100010 1;
2. *Écrire les fonctions de bas niveau pour écrire et lire des bits et octets dans le fichier ;*
3. *Écrire la bibliothèque et la tester.*