

## **Exercice 1 : Nombres entiers**

Pour commencer, on rappelle les procédures permettant de coder et de décoder en  $\text{UNSIGNED-}k$ , CA2 et  $\text{SIGNED-}k$ .

### **Encoder un nombre décimal positif en $\text{UNSIGNED-}k$**

1. Convertir ce nombre décimal en base 2.
2. Si le nombre de bits obtenu dépasse  $k$ ; Rejeter le nombre : il n'est pas encodable en  $\text{UNSIGNED-}k$  car trop grand.
3. Si le nombre de bits est plus petit que  $k$ ; Ajouter des 0 à gauche jusqu'à avoir  $k$  bits.

### **Encoder une suite de bits en CA2**

1. Inverser la valeur de chaque bit (0 devient 1 et 1 devient 0)
2. On ajoute 1

### **Décoder un code CA2**

1. Inverser la valeur de chaque bit (0 devient 1 et 1 devient 0)
2. On ajoute 1

### **Encoder un nombre décimal en $\text{SIGNED-}k$**

- Si ce nombre est positif :
1. Retourner  $\text{UNSIGNED-}k$  de ce nombre
- Si ce nombre est négatif :
1. Prendre la valeur absolue de ce nombre
  2. Convertir le résultat en binaire sur  $k$  bits
  3. Retourner CA2 de la séquence binaire précédemment trouvée

### **Décoder un code en $\text{SIGNED-}k$** On regarde le premier bit de ce code

- Si ce bit est 0 :
1. Convertir le code binaire en décimal
- Si ce bit est 1 :
1. Décoder en suivant le décodage de CA2
  2. Convertir le code obtenu en décimal
  3. Multiplier par  $-1$

#### **Q1.** Codez les entiers suivants en utilisant le codage signed-8 :

- 25
- 42
- -13
- -25
- -100

#### **Q2.** Exprimez les résultats précédents en hexadécimal.

#### **Q3.** Les octets suivants sont les résultats, exprimés en hexadécimal, de conversion utilisant signed-8. À quels entiers correspondent-ils ?

- 0x4A
- 0x1C
- 0xDB
- 0xF5
- 0x85

## Exercice 2 : Base 64

Le codage base64 est un codage permettant de transformer toute donnée binaire en une donnée n'utilisant que 64 caractères ASCII. Il permet, par exemple, de transmettre n'importe quelle donnée binaire par les protocoles de communication n'autorisant que les caractères ASCII (comme le courrier électronique).

Le principe de ce codage consiste à découper la donnée binaire en tranches de six bits, appelées *sextets*, et d'associer à chaque sextet un caractère choisi parmi les 26 lettres latines capitales et minuscules, les 10 chiffres arabes et les deux caractères + et /.

La table de codage est la suivante :

Sextet (déc.)	Code						
000000 (0)	A	010000 (16)	Q	100000 (32)	g	110000 (48)	w
000001 (1)	B	010001 (17)	R	100001 (33)	h	110001 (49)	x
000010 (2)	C	010010 (18)	S	100010 (34)	i	110010 (50)	y
000011 (3)	D	010011 (19)	T	100011 (35)	j	110011 (51)	z
000100 (4)	E	010100 (20)	U	100100 (36)	k	110100 (52)	0
000101 (5)	F	010101 (21)	V	100101 (37)	l	110101 (53)	1
000110 (6)	G	010110 (22)	W	100110 (38)	m	110110 (54)	2
000111 (7)	H	010111 (23)	X	100111 (39)	n	110111 (55)	3
001000 (8)	I	011000 (24)	Y	101000 (40)	o	111000 (56)	4
001001 (9)	J	011001 (25)	Z	101001 (41)	p	111001 (57)	5
001010 (10)	K	011010 (26)	a	101010 (42)	q	111010 (58)	6
001011 (11)	L	011011 (27)	b	101011 (43)	r	111011 (59)	7
001100 (12)	M	011100 (28)	c	101100 (44)	s	111100 (60)	8
001101 (13)	N	011101 (29)	d	101101 (45)	t	111101 (61)	9
001110 (14)	O	011110 (30)	e	101110 (46)	u	111110 (62)	+
001111 (15)	P	011111 (31)	f	101111 (47)	v	111111 (63)	/

### Encoder une donnée binaire en BASE64

1. Écrivez votre donnée en binaire.
2. Découper la donnée binaire par bloc de 6 en partant de la gauche (compléter avec des zeros à droite si nécessaire).
3. Remplacer chaque bloc par le caractère associé dans la table.

### Décodé un texte BASE64

1. Remplacer chaque bloc par la séquence de 6 bits associée dans la table
2. Si le nombre de bits n'est pas un multiple de 8, supprimer les 0 les plus à droite jusqu'à avoir un nombre de bits multiple de 8.

**Q1.** Quel est le codage en base64 des 3 octets 0x20 0x42 0x08 ?

**Q2.** Quels sont, exprimés en binaire et en hexadécimal, les octets codés en base 64 par ZMOpY29kYWdl ?

**Q3.** Cette chaîne représente un mot lui-même codé en UTF-8. Quel est ce mot ?