

Université de Montréal

Exercice noté 1

par  
Ariane Brucher, 20144709  
Héloïse Tanguay, 20161242

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Travail présenté à Mme Alena Tsikhanovich  
dans le cadre du cours IFT1016B  
Programmation 1

24 septembre 2019

## 1. Conversions de base

- **Déterminer la puissance de chaque chiffre pour un nombre de 5 chiffres en base 7.**

Pour un nombre donné ABCDE, la puissance correspond au poids associé à la position du nombre. Ainsi, A a une puissance de 4, B une puissance de 3, C de 2, D de 1 et E de 0.

$7^4$	$7^3$	$7^2$	$7^1$	$7^0$
A	B	C	D	E

- **Convertir le nombre  $2AA3_{16}$  en décimal.**

$$2AA3_{16} = (2 \cdot 16^3 + 10 \cdot 16^2 + 10 \cdot 16^1 + 3 \cdot 16^0)_{10} = 10915_{10}$$

- **Convertir le nombre  $4B_{16}$**

- **d'hexadécimal à décimal :**

$$4B_{16} = (4 \cdot 16^1 + 11 \cdot 16^0)_{10} = 75_{10}$$

- **d'hexadécimal à binaire :**

Méthode de division successive pour convertir en base 2

$$75/2 = 37 \text{ reste } 1$$

$$37/2 = 18 \text{ reste } 1$$

$$18/2 = 9 \text{ reste } 0$$

$$9/2 = 4 \text{ reste } 1$$

$$4/2 = 2 \text{ reste } 0$$

$$2/2 = 1 \text{ reste } 0$$

$$1/2 = 0 \text{ reste } 1 \rightarrow \text{fin de la division successive}$$

$$\text{Donc } 4B_{16} = 1001011_2$$

- **d'hexadécimal à octal :**

Méthode de division successive pour convertir en base 8

$$75/8 = 9 \text{ reste } 3$$

$$9/8 = 1 \text{ reste } 1$$

$$1/8 = 0 \text{ reste } 1 \rightarrow \text{fin de la division successive}$$

$$\text{Donc } 4B_{16} = 113_8$$

- **Encoder l'entier  $1011_{10}$  avec la notation hexadécimale de JavaScript :**

Méthode de division successive pour convertir en base 16

$$1011/16=63 \text{ reste } 3$$

$$63/16=3 \text{ reste } 15 \text{ (ou } F_{16})$$

$$3/16=0 \text{ reste } 3 \rightarrow \text{fin de la division successive}$$

$$\text{Donc } 1011_{10}=3F3_{16}=0X3F3$$

- **Valeur de 0XEE en JavaScript :**

$$0XEE=EE_{16}=(14*16^1+14*16^0)_{10}=238_{10}$$

## 2. Représenter un nombre $17_{10}$ selon la convention non signée sur 5 bits.

Convention non signée (trouver par division successive pour convertir en base 2)

$$17/2=8 \text{ reste } 1$$

$$8/2=4 \text{ reste } 0$$

$$4/2=2 \text{ reste } 0$$

$$2/2=1 \text{ reste } 0$$

$$1/2=0 \text{ reste } 1 \rightarrow \text{fin de la division successive}$$

Donc,  $17_{10}=10001_2$  en encodage non signé sur 5 bits

## 3. Quelles valeurs sont encodées par la convention complément à 2 sur 5 bits par les chaînes binaires suivantes :

- $01101_2$  est un nombre positif, donc nous utilisons la représentation binaire non signée sur  $n$  bits :

$$01101_2=(0*2^4+1*2^3+1*2^2+0*2^1+1*2^0)_{10}=13_{10}$$

- $10011_2$  est un nombre négatif, donc nous utilisons l'encodage binaire signé en complément à 2 :

$$10011_2=(1*(-2^4)+0*2^3+0*2^2+1*2^1+1*2^0)_{10}=-13_{10}$$

## 4. L'anatomie d'un nombre à virgule flottante précision double IEEE 754 (64 bits) :

Un nombre à virgule flottante précision double IEEE 754 a un total de 64 bits et est composé de nombres binaires (base 2). Tout d'abord, le premier bit indique le signe ( $s$ ) du nombre : 0 correspond à un chiffre positif et 1 indique que le chiffre est négatif. Les 11 bits suivants, désignés par  $e$ , représentent la valeur de l'exposant  $e_1$ , à laquelle on additionne  $1023_{10}$  ( $e = e_1 + 1023_{10}$ ). Puis, les 52 derniers bits sont le champ fraction ( $f$ ). Les chiffres qui

composent le champ fraction sont les décimales du nombre binaire trouvé lors de la conversion du nombre donné initialement au nombre binaire, lorsque la virgule suit le premier 1 du nombre binaire.

### Quel est l'encodage en précision double IEEE 754 (64 bits) des nombres point flottant...

- **$3,15_{10}$**

1) Entiers :

base 10  $\rightarrow$  base 2

$$3_{10} = 11_2$$

Fractions :

$0,15_{10} \rightarrow$  base 2

$$0,15_{10} = 0,0010\overline{0110}_2$$

Méthode de multiplication successive :

$$0,15 \times 2 = 0,3$$

$$0,3 \times 2 = 0,6$$

$$0,6 \times 2 = 1,2$$

$$0,2 \times 2 = 0,4$$

$$0,4 \times 2 = 0,8$$

$$0,8 \times 2 = 1,6$$

$$0,6 \times 2 = 1,2$$

$$0,2 \times 2 = 0,4$$

$$0,4 \times 2 = 0,8$$

...

$$2) 3,15_{10} = 11,0010\overline{0110}_2$$

$$3,15_{10} = 1,1001\overline{0011}_2 \times 2^1$$

$$3) e_1 = 1$$

$$e = 1 + 1023 = 1024_{10} = 10000000000_2$$

$$4) \underbrace{0}_s \underbrace{1000000000}_e \underbrace{1001\overline{0011}}_f \text{ (le champ } f \text{ arrondit le nombre lorsqu'il atteint 52 bits)}$$

- **$-4_{10}$**

Puisque  $-4_{10}$  est un nombre négatif, le nombre dans le champ  $s$  sera 1.

2) Entiers :

base 10  $\rightarrow$  base 2

$$4_{10} = 100_2$$

$$2) 4_{10} = 100, \bar{0}_2$$

$$4_{10} = 1, \bar{0}_2 * 2^2$$

$$3) e_1 = 2$$

$$e = 2 + 1023 = 1025_{10} = 10000000001_2$$

$$4) \underbrace{1}_s \underbrace{10000000001}_e \underbrace{\bar{0}}_f \text{ (dans le champ } f, \text{ dans ce cas, il y aura 52 zéros)}$$

Fractions :

$0_{10} \rightarrow$  base 2

$$0_{10} = 0, \bar{0}_2$$

**5. Plus petite expression JavaScript dont la valeur est 9 :**

$$10 - (2 + 3) + 4$$

**6. Programme de calcul de paiements mensuels d'un prêt : voir le fichier joint à la remise.**

```
var p=200000;
```

```
var n=360;
```

```
var t=5.5/100;
```

```
var x=t/(12*100);
```

```
var y=Math.pow((x+1),n);
```

```
var m=(p*y*x)/(y-1);
```

```
print(m);
```