



# Architecture des ordinateurs

Département Informatique

Erwan LEBAILLY — Vilavane LY — Vincent TRÉLAT — Benjamin ZHU

*21 février 2022*

\*\*\*

# Table des matières

|          |                      |          |
|----------|----------------------|----------|
| <b>1</b> | <b>Exercices</b>     | <b>2</b> |
| 1.1      | Exercice 1 . . . . . | 2        |
| 1.2      | Exercice 2 . . . . . | 2        |
| 1.3      | Exercice 3 . . . . . | 2        |
| 1.4      | Exercice 4 . . . . . | 2        |
| 1.5      | Exercice 5 . . . . . | 3        |
| 1.6      | Exercice 6 . . . . . | 3        |
| 1.7      | Exercice 7 . . . . . | 3        |

# 1 Exercices

## 1.1 Exercice 1

Avec la convention  $0 \leftrightarrow \text{faux}$  et  $1 \leftrightarrow \text{vrai}$ ,  $0 \wedge 1 = \text{faux}$ .

## 1.2 Exercice 2

On donne la table de  $c_0$  :

|                      |   |   |
|----------------------|---|---|
| $a_0 \backslash b_0$ | 0 | 1 |
| 0                    | 0 | 1 |
| 1                    | 1 | 0 |

On peut interpréter cette table comme la table de vérité du "ou exclusif", le *xor*. Ainsi,  $c_0$  coïncide avec  $a_0 \oplus b_0 = (a_0 \vee b_0) \wedge (\neg(a_0 \wedge b_0))$ .

## 1.3 Exercice 3

- Montrer que xor est associatif et commutatif puis recopier calcul
- inclure schéma

## 1.4 Exercice 4

- On écrit le code suivant :

```
1 int main()  
2 {  
3     printf("Sizeof int: %lu octets\n", sizeof(int));  
4     printf("Sizeof short: %lu octets\n", sizeof(short));  
5     printf("Sizeof char: %lu octets\n", sizeof(char));  
6     return 0;  
7 }
```

La sortie est la suivante :

```
Sizeof int: 4 octets  
Sizeof short: 2 octets  
Sizeof char: 1 octets
```

- On écrit le code suivant :

```

1 int main()
2 {
3     int a = pow(2, 31);
4     int b = pow(2, 31);
5     int c = a + b;
6     printf("%d\n", c);
7     return 0;
8 }

```

La sortie affiche 0, ce qui correspond bien à  $2^{32} \bmod (2^{32})$

## 1.5 Exercice 5

On donne ci-dessous l'écriture binaire sur 4 et 8 bits de 0, 1, -1 et -2 :

| $x$  | 4 bits | 8 bits    |
|------|--------|-----------|
| 0 :  | 0000   | 0000 0000 |
| 1 :  | 0001   | 0000 0001 |
| -1 : | 1111   | 1111 1111 |
| -2 : | 1110   | 1111 1110 |

## 1.6 Exercice 6

- $m_1 = 0001$  et  $m_{-1} = 1001$ .
- En abusant de la notation + pour des mots :  $m_0 = m_1 + m_{-1} = 1010$ .
- En suivant la règle de signes, 1010 est l'encodage de -2.

## 1.7 Exercice 7