IN302

TD03

→ Position du bit à 1 le plus à gauche

1)
```
char_1er_bit_a_un(uint32_t M){

  char pos = 0;
  while (M>0){
  M = M>>1;              au max, 32 itérations
   pos++;
   }
  return pos;
}
```

2)
```
char 1er_bit_a_un_dich(uint32_t M)
{
  char pos = 0;   for(i=0; i<=4; i++){
  char i;           if(M and masque[i]){
                      M = M>>shift[i];
                      pos += shift[i];
                    }} return pos;}
shift[5]={16,8,4,2,1}
masque[5]={0xFFFF0000,0xFF00,0xF0,0b1100,0b10}
```

→ code 2 parmi 5

1) ||,,, ,,,|| ,,|,| ,,,||

1 9 8 9

2) $d(C_6,C_9)=d(10001,00011)=2$

$d(C_5,C_6)=d(00110,10001)=4$

3) c_i et c_j ont 2 bits à 1.

* 1 bit à un partagé par c_i et c_j et 1 bit différent.

$n_i < m_i$ positions bits à 1 de c_i
$n_j < m_j$ " " " de c_j
$\Rightarrow m_i \neq m_j$. Alors distance Hamming est 2.

* aucun bit à 1 partagé, distance Hamming est 4. Donc $d_c = \min(2, 4) = 2$

4) $e_d = 2 - 1 = 1$ $e_c = 0$

→ code avec bit de parité

1) 01010 et 11000

2) 2 mots consécutifs du code : 00000 et 10001,
$d_H(00000, 10001) = 2$, donc $d_c \leq 2$

Soit 2 mots de 4 bits distincts x et y,

- Si $d_H(x, y) \geqslant 2$ alors $d_H(\text{parité}(x), \text{parité}(y)) \geqslant 2$
- Si $d_H(x, y) = 1$ alors $\text{parité}(x) \neq \text{parité}(y)$

donc $d_H(\text{parité}(x), \text{parité}(y)) = 2$
Dans tout les cas $d_H \geqslant 2$ donc $d_c = 2$

3) $e_d = 2 - 1 = 1$. aucune peut être corrigé

→ code à répétition

1) 2 mots dans image code, 000 et 111.
la distance est donc 3.

2) ed = 3-1 = 2 ec = 1 erreur peut être corrigé

3) 1er message : 1 1 1 0 0 0 1 1 1
pas d'erreur
2ème message : 1 1 0 1 0 0 0 1 0 1 1 1 0 1 1
3 erreurs × ✓ × ×

→ Code de Hamming (7,4)

1) a3 a2 a1 a0 → p1 p2 a0 p3 a1 a2 a3
p1 = a0, a1, a3 p2 = a0 a2 a3 p3 = a1 a2 a3

2 bits de parités changent pour a0, a1, a2
et 3 bits de parité pour a3.

2) distance Hamming 2 bits parités et 1
pour celui qui change, donc 3.
2 erreurs et corriger qu'1 seule.

3) 000111 p1 = 0, p2 = 0, a0 = 0, a1 = 1, p3 = 1,
a2 = 1, a3 = 1

p1 = 011 p2 = 011 p3 = 111 correct
→ p1 = 0 → p2 = 0 → p3 = 1

1010010
p1 = 100 p2 = 110 p3 = 010
→ p1 = 1 → p2 = 0 → p3 = 1 p3, p2 incorrects
changer a2 → p2 = 1 et p3 = 0

1100 1110

p1 = 0 1 0 p2 = 0 1 0 p3 = 1 1 0

→ p1 = 1 → p2 = 1 → p3 = 0,

p3 incorrect, erreur bit parité

00 1 1 1 1 1

p1 et p2 incorrects, changer a0.