#1 a)

on choisit H et G

$$H = \left(\begin{array}{cccc|ccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right)_{3\times 7}$$

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ & & & & 1 & 1 & 1 \end{pmatrix}$$

$u = (0,0,0,0)$, $uG = v = (0,0,0,0,0,0,0)$.

$u = (0,0,1,0)$, $uG = v = (0,0,1,0,0,1,1)$

$u = (0,1,1,1)$, $uG = v = (0,1,1,1,0,0,1)$.

b) $v' = (1,1,1,1,1,1,1)_{1\times 7}$

on calcul $Hv'^T = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, pas d'erreur

$\Rightarrow v' = v$, et les 4 ières coordo forment le message de départ : $u = (1,1,1,1)$.

$v' = (1,0,1,1,1,1,1)$.

$Hv'^T = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow$ erreur, au 2^e bit

$\Rightarrow v = (1,1,1,1,1,1,1)$.

$\Rightarrow u = (1,1,1,1)$.

$v' = (0,0,0,0,1,1,1)$, $Hv'^T = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \Rightarrow 4^e$ bit erreur

$\Rightarrow v = (0,0,0,1,1,1,1) \Rightarrow u = (0,0,0,1)$.

$v' = (1,1,1,1,0,0,0)$ $Hv'^T = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \Rightarrow v = (1,1,1,0,0,0,0)$

$\Rightarrow u = (1,1,1,0)$.

#2 a)

$w = (1,0,1,0,1,0,1,0,1,0,1,0,1,0,1)$

on calcul $Hw^T = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

oui, il y a une erreur à la 1ère composante de w :

$v = (0,0,1,0,1,0,1,0,1,0,1,0,1,0,1)$

$\Rightarrow u = (0,0,1,0,1,0,1,0,1,0,1)$.

b) $C(2^k - 1, 2^k - k - 1)$

$2^k - k - 1 \longrightarrow$ longueur mot original

$2^k - 1 \longrightarrow$ longueur mot transmis.

on veut que $1{,}1(2^k - k - 1) \geq 2^k - 1$

$\Leftrightarrow 0{,}1 \cdot 2^k - 0{,}1 \geq 1{,}1 \cdot k$.

$\Leftrightarrow 2^k - 1 \geq 11k$

vrai pour $k \geq 7$

$\boxed{\#7}$ a)

$H = (M_{k\times n-k} \mid I_{k\times k})$, $G = (I_{n-k\times n-k} \mid M^T_{n-k\times k})$

regardons si les lignes de G sont orthogonales à celles de H;

i^e ligne de H fois l^e ligne de G, $1 \leq i \leq k$, $1 \leq l \leq n-k$

$$\sum_{j=1}^{n} H_{ij} G_{lj} = \sum_{j=1}^{n-k} H_{ij} G_{lj} + \sum_{j=n-k+1}^{n} H_{ij} G_{lj}$$

$$= M_{il} + M^T_{li} = 2 M_{il} \equiv 0 \pmod 2.$$

alors les lignes sont bien orthogonales.

Aussi, G est de rang plein par l'identité de ses $n-k^e$ premières composantes b)

$$H_{4\times 15} = \left(\begin{array}{ccccccccccc|cccc} \multicolumn{11}{c|}{M} & \multicolumn{4}{c}{I} \\ 1&1&1&1&0&1&1&0&1&0&0 & 1&0&0&0 \\ 1&1&1&0&1&1&0&1&0&1&0 & 0&1&0&0 \\ 1&1&0&1&1&0&1&1&0&0&1 & 0&0&1&0 \\ 1&0&1&1&1&0&0&0&1&1&1 & 0&0&0&1 \end{array}\right)$$

$$G_{11\times 15} = \left(\begin{array}{cccccc|cccc} \multicolumn{6}{c|}{I} & \multicolumn{4}{c}{M^T} \\ 1&0&0&0&&& 1&1&1&1 \\ 0&1&0&0&&& 1&1&1&0 \\ 0&0&1&&&& 1&1&0&1 \\ 0&0&0&&\cdots&& 1&0&1&0 \\ 0&0&0&&&& 0&1&1&1 \\ 0&0&0&&&& 1&1&0&0 \\ 0&0&0&&&& \vdots&&& \\ 0&0&0&&&& &&& \\ 0&0&0&&&& &&& \\ 0&0&0&&&& &&& \\ 0&0&0&&&& &&& \end{array}\right)$$

c) $v' = (1,1,1,1,1,0,0,0,0,0,1,1,1,1,1)$

on calcul $H v'^T = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow 6^e$ ligne, pas bonne.

$\Rightarrow v = (1,1,\underline{1},1,1,\underline{1},0,0,0,0,1,1,1,1,1)$

$\Rightarrow u = (1,1,\underline{1},1,1,\underline{1},0,0,0,0,1)$

$\boxed{\#8}$

- avoir 2 bits fautifs ; $p \cdot p = p^2$
- avoir le reste des bits non-fautifs ; $(1-p)^5$.
- choisir la place des bits fautifs ; $\binom{7}{2}$

alors, proba d'avoir 2 fautifs est :

$p^2 \cdot (1-p)^5 \cdot \binom{7}{2}$.

$\approx 0{,}0000208952$

b) P(avoir plus d'une erreur)

$= 1 - P(\text{avoir 1 erreur ou moins})$

$= 1 - (P(\text{avoir aucune erreur}) + P(\text{avoir 1 erreur}))$

$P(\text{aucune erreur}) = (1-P)^7 \approx 0,993021$

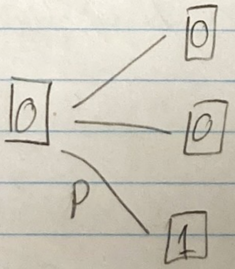
$P(\text{1 erreur}) = P \cdot (1-P)^6 \cdot \binom{7}{1} = 0,00696$

$\Rightarrow P(\text{avoir plus d'une erreur})$

$\approx 1 - 0,993021 - 0,00696$

$= 0,0000209301$

c)



Pour décoder correctement, il faut qu'il y ait au plus une erreur ;

$P(\text{aucune erreur}) = (1-P)^3 \approx 0,997003$

$P(\text{1 seule erreur}) = P \cdot (1-P)^2 \cdot 3 \approx 0,002994$

$\Rightarrow P(\text{décoder}) \approx 0,999997$

d) P(4 bits décodé correct)

$= (0,999997)^4 = 0,999988$

Proba que le code simple fonctionne.

Proba que Hamming fonctionne

= 1 - P(avoir plus d'une erreur de transmission)
(b)

$= 1 - 0,0000209301 = 0,999979$