Comment transmettre des données et fiabiliser la transmission ?

Activités avec prof	Activités autonomes débranchées	Activités autonomes branchées
Démo d'une « transmission de données » avec pertes.		
	Découverte du bit de parité	
		Etablir une connexion entre un serveur et un client en python
	Codage, décodage, base 2, base 10	
	Lecture d'une table de caractères (ASCII)	
		Codage, décodage, base 2, base 10 en python
		Lecture d'une documentation python. Ecrire une fonction permettant de convertir un entier en caractère. Ecrire une fonction permettant de convertir un entier en caractère.
Cours parité		
Evaluation QCM – base 2 – base 10 – caractère - parité		
	Ecrire un algorithme de calcul de la parité	
	Ecrire un algorithme de détection d'erreur	
	Ecrire un algorithme d'extraction de la donnée (7 bits)	
		Ecrire et tester une fonction de calcul de la parité
		Ecrire et tester une fonction de détection d'erreur
		Ecrire une fonction et tester d'extraction de la donnée (7 bits)
		L'enseignant met à disposition un serveur python. Ce serveur délivre du texte codé avec une parité paire et « simulant » des erreurs lors de la transmission en fournissant le texte avec des erreurs aléatoires.

		Les élèves devront se connecter à ce serveur, récupérer les données transmises, afficher ses données, détecter informatiquement les erreurs, compter le nombre d'erreurs.
		Les élèves créent eux même le serveur de texte « bruité »
	Calculs divers (Jerome)	

Exemple de codes pour la partie réception sans la partie réseau

```
def codageBin(n):
    binaire=[]
    for i in range(8):
        quotient=n//2
        reste=n%2
        binaire.append(reste);
        n=quotient
    return binaire

def decodageBin(b):
    return b[7]*2**7+b[6]*2**6+b[5]*2**5+b[4]*2**4+b[3]*2**3+b[2]*2**2+b[1]*2**1++b[0]*2**0

def testParite(l_bin):
    nbre1=0
    for i in range(8):
        if l_bin[i]==1:
            nbre1+=1
    if nbre1%2==0:
        return True
    return False

def convCharToInt(caract):
    return ord(caract)

def convIntToChar(nbre):
    return chr(nbre)

def supprParite(nbre):
    if nbre>128:
        return nbre-128
    else:
        return nbre

def ajoutParite(nbre):
    codeBin=codageBin(nbre)
    if not testParite(codageBin(nbre)):
        codeBin[7]=1
    return decodageBin(codeBin)
```

```
caractere ='B'
integer=convCharToInt(caractere)                66
print(integer)                                  66
integer=ajoutParite(integer)                    B
print(integer)
caractParite=convIntToChar(integer)
print(caractParite)

integer=convCharToInt(caractParite)
print(integer)                                  66
integer=supprParite(integer)                    66
print(integer)                                  B
caractere=convIntToChar(integer)
print(caractere)

print("------------------------")                ------------------------

caractere ='C'                                  67
integer=convCharToInt(caractere)                195
print(integer)                                  Ã
integer=ajoutParite(integer)
print(integer)
caractParite=convIntToChar(integer)
print(caractParite)

integer=convCharToInt(caractParite)             195
print(integer)                                  67
integer=supprParite(integer)                    C
print(integer)
caractere=convIntToChar(integer)
print(caractere)

print("------------------------")                ------------------------
print(supprParite(120))                         120
print(supprParite(130))                         2
print("------------------------")                ------------------------
b=print(codageBin(120))                         [0, 0, 0, 1, 1, 1, 1, 0]
b=print(codageBin(130))                         [0, 1, 0, 0, 0, 0, 0, 1]
print("------------------------")                ------------------------
print(testParite(codageBin(120)))               True
print(testParite(codageBin(130)))               True
print("------------------------")                ------------------------
print(ajoutParite(64))                          192
print(ajoutParite(65))                          65
print("------------------------")                ------------------------
```