

# DEVOIR SURVEILLÉ D'INFORMATIQUE 1

## CI 1 : ARCHITECTURE MATÉRIELLE ET LOGICIELLE

## CI 2 : ALGORITHMIQUE ET PROGRAMMATION

### 1 Codage des nombres

Pour tout ce devoir, on dispose d'une machine dont le codage est limité à 8 bits.

#### 1.1 Capacités de l'espace machine

##### Question 1

Quel est le nombre maximum d'entiers qu'il est possible de coder ? Donner le nombre maximal et le nombre minimal dans les systèmes décimal, binaire et hexadécimal.

##### Question 2

Quel est le nombre maximum d'entiers relatifs qu'il est possible de coder ? Donner le nombre minimal et le nombre maximal dans le système décimal.

#### 1.2 Conversions

Dans cette partie, les nombres sont tous des entiers relatifs codés en complément à 2.

##### Question 3

Convertir le nombre 83 dans le système binaire et dans le système hexadécimal.

##### Question 4

Faire la somme  $83 + 200$ . Donner le résultat en binaire.

##### Question 5

Réaliser l'opération  $24 - 83$ . Donner le résultat en binaire puis en décimal.

##### Question 6

Convertir le nombre  $(A3)_{16, \mathbb{Z}}$ . Coder ce nombre dans le système décimal.

#### 1.3 Algorithmique et programmation

Le but de cette partie est de réaliser un programme permettant de réaliser le codage d'un nombre entier relatif en utilisant le codage en complément à 2.

Une chaîne de caractère se comporte comme un liste. En effet prenons par exemple la chaîne de caractères exemple :

```
>>> chaine = "exemple"
>>> print(chaine)
exemple
>>> len(chaine) # Retourne le nombre de caractères de la chaîne : il y a 7 caractères dans le mot exemple
7
>>> print(chaine[0]) # Affiche le premier e
'e'
>>> print(chaine[6]) # Affiche le dernier e
'e'
>>> for i in range(0,2,1) : # Pour i allant de 0 (inclus) à 2 (exclus) par pas de 1, faire :
    print(str(i)+' : '+''+chaine[i])
0 : e
1 : x
>>> chaine = chaine+'s'
```

python

```
>>> print(chaine)
      exemples
>>> chaine = "Les " + chaine
>>> print(chaine)
      Les exemples
```

16  
17  
18  
19  
20

### 1.3.1 Conversion d'un nombre décimal en binaire

On donne l'extrait de programme suivant permettant de convertir un nombre entier positif `nb` en chaîne de caractères `res` dont le contenu est le nombre en binaire.

python

```
nb = 10
dividende = nb
diviseur = 2
resultat = ""
quotient = -nb

while quotient != 0 :
    quotient = int(dividende / diviseur)
    reste = dividende - diviseur * quotient
    dividende = quotient
    resultat = resultat + str(reste)

res = ""
for i in range(len(resultat)-1, -1, -1):
    res = res + resultat[i]
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

#### Question 7

Quel est le type des variables `dividende` et `resultat`.

#### Question 8

Expliquer la ligne 7. Justifier ce choix.

#### Question 9

On cherche à analyser l'évolution des variables lors du parcours de la boucle `while`. Remplir les champs suivants.

État des variables à la ligne 6 :

- `nb` : .....
- `dividende` : .....
- `diviseur` : .....
- `resultat` : .....
- `quotient` : .....

- Première itération de la boucle `while` – État des variables après exécution de la ligne 11 :

- `quotient` : .....
- `reste` : .....
- `dividende` : .....
- `resultat` : .....

- Seconde itération de la boucle `while` – État des variables après exécution de la ligne 11 :

- `quotient` : .....
- `reste` : .....
- `dividende` : .....
- `resultat` : .....

- Troisième itération de la boucle `while` – État des variables après exécution de la ligne 11 :

- `quotient` : .....
- `reste` : .....

- dividende : .....
- resultat : .....
- Quatrième itération de la boucle *while* – État des variables après exécution de la ligne 11 :
  - quotient : .....
  - reste : .....
  - dividende : .....
  - resultat : .....

### Question 10

Parmi les lignes 8, 9 et 10, réaliser des modifications qui permettent de mieux utiliser les opérations disponibles en Python.

### Question 11

Après exécution de la liste que contient la variable *resultat* ? Combien vaut *len(resultat)* ?

### Question 12

On cherche à analyser l'évolution des variables lors du parcours de la boucle *for*. Remplir les champs suivants.

Valeur de i :	
Valeur de res :	
Valeur de i :	
Valeur de res :	
Valeur de i :	
Valeur de res :	
Valeur de i :	
Valeur de res :	
Valeur de i :	
Valeur de res :	

### 1.3.2 Programme mystère

On cherche à convertir le nombre  $(-10)_{10}$  en base 2. Le système utilisé utilise un codage sur 8 bits. La conversion du nombre  $(10)_{10}$  en binaire est  $(1010)_2$ .

On donne cette partie de programme.



```
res_cv = "1010"
nb_bits = 8
while(len(res_cv)!=nb_bits):
    res_cv = "0"+res_cv
```

1  
2  
3  
4

### Question 13

Quel est le but du programme précédent ? Que contient `res_cv` après l'exécution du code ?

### 1.3.3 Inversion des bits

On cherche maintenant à inverser les bits d'une séquence.



```
res_cv = "1010"
res_inv = ""
for i in range(len(res_cv)):
    if res_cv[i]=="0":
        res_inv=res_inv+"0"
    else :
        res_inv=res_inv+"1"
```

1  
2  
3  
4  
5  
6  
7

### Question 14

Que contient `res_cv` après l'exécution de la boucle ?

### Question 15

Si le résultat obtenu n'est pas le résultat attendu, comment modifier la séquence précédente ?

**Fin des questions**

### 1.3.4 Pour information

A titre d'information, voici une séquence de programme permettant d'ajouter 1 à un nombre codé en binaire.



```
# On ajoute +1
# Initialisation
retenue="1"
res=""
for i in range(len(res_inv)-1,-1,-1):
    if retenue=="0" and res_inv[i]=="0":
        retenue="0"
        res = "0"+res
    elif retenue=="0" and res_inv[i]=="1":
        retenue = "0"
        res = "1"+res
    elif retenue=="1" and res_inv[i]=="0":
        retenue = "0"
        res = "1"+res
    elif retenue=="1" and res_inv[i]=="1":
        retenue = "1"
        res = "0"+res
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17