IFT-1015

Démo 2

Exercice 1

Base 10	Base 16	Base 8	Base 2
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
10	a	12	1010
11	b	13	1011
12	c	14	1100
13	d	15	1101
14	e	16	1110
15	f	17	1111
16	10	20	10000
17	11	21	10001
18	12	22	10010
19	13	23	10011
20	14	24	10100

Exercice 2

$64_{16} = 16^1 \cdot 6 + 16^0 \cdot 4 = 100_{10}$

$100_8 = 8^2 \cdot 1 + 8^1 \cdot 0 + 8^0 \cdot 0 = 64_{10}$

$2988_{10} = 16^2 \cdot 11 + 16^1 \cdot 10 + 16^0 \cdot 12 = bac_{16} = 0\text{xbac}$

$0\text{xff} = ff_{16} = 16^1 \cdot 15 + 16^0 \cdot 15 = 255_{10}$

Exercice 3

Convertir en binaire(signé), inverser les bits, convertir en décimal
$\sim 12_{10} \rightarrow \sim 01100_2 \rightarrow 10011_2 \rightarrow -13_{10}$

Convertir en binaire, appliquer le "et" logique, convertir en décimal
3_{10} & $5_{10} \rightarrow 0011_2$ & $0101_2 \rightarrow 0001_2 \rightarrow 1_{10}$

Convertir en binaire, appliquer le "ou" logique, convertir en décimal
$3_{10} \mid 5_{10} \rightarrow 0011_2 \mid 0101_2 \rightarrow 0111_2 \rightarrow 7_{10}$

Convertir en binaire, appliquer le "exclusive or" logique, convertir en décimal
3_{10} ˆ$5_{10} \rightarrow 0011_2$ ˆ$0101_2 \rightarrow 0110_2 \rightarrow 6_{10}$

Convertir en binaire le nombre de gauche, appliquer le décalage logique vers la gauche, convertir en décimal
$3_{10} << 4_{10} \rightarrow 0011_2 << 4_{10} \rightarrow 00110000_2 \rightarrow 48_{10}$

Convertir en binaire le nombre de gauche, appliquer le décalage logique vers la droite, convertir en décimal
$100_{10} >> 2_{10} \rightarrow 01100100_2 >> 2_{10} \rightarrow 0011001_2 \rightarrow 25_{10}$

Exercice 4

nombre	s	e	f
0.0	0	00000000000	0000000000000000000000000000000000000000000000000000
7.0	0	10000000001	1100000000000000000000000000000000000000000000000000
-2.5	1	10000000000	0100000000000000000000000000000000000000000000000000
1.0	0	01111111111	0000000000000000000000000000000000000000000000000000
2.0	0	10000000000	0000000000000000000000000000000000000000000000000000

Exercice 5

$\sim$ x == -1-x

Exercice 6

x%2 == x&1

Exercice 7

```
a = 2   # partie 1
b = 1   # partie 2
c = 5   # partie 3

facteur = 100/(a+b+c)

print(a*facteur)
print(b*facteur)
print(c*facteur)
```

Exercice 8

```

# Ce programme permet de trouver le jour de la semaine
# pour une certaine date.  Le résultat imprimé indique
# le jour de la semaine, avec 1=dimanche, 2=lundi, etc.

annee = 2020    # annee de la date
mois = 9        # mois de la date
quant = 8       # quantieme de la date

# Calculer l'ajustement des années bisextiles
nbMois = annee*12 + mois - 3  # nombres de mois écoulés
nbAns = nbMois // 12          # nombres d'années écoulées
nb4Ans = nbAns // 4           # nombre périodes de 4 ans écoulées
nb100Ans = nbAns // 100       # nombre périodes de 100 ans écoulées
nb400Ans = nbAns // 400       # nombre périodes de 400 ans écoulées
ajustBisext = (mois+9)//12*4 + nbAns + nb4Ans - nb100Ans + nb400Ans

# Calculer le jour de la semaine
jour = (23*mois//9 + ajustBisext + quant + 5) % 7 + 1

print(jour)
```