Devoir surveillé n°4 – NSI
Correction

Exercice 1 (5 points)

1. La partie entière de $\frac{19}{7}$ est $\frac{14}{7} = 2 = (10)_2$. La partie décimale est $\frac{5}{7}$. Par suite,

$$\begin{aligned} \frac{5}{7} \times 2 &= \frac{10}{7} = \frac{7}{7} + \frac{3}{7} && \to 1 \\ \frac{3}{7} \times 2 &= \frac{6}{7} && \to 0 \\ \frac{6}{7} \times 2 &= \frac{12}{7} = \frac{7}{7} + \frac{5}{7} && \to 1 \end{aligned}$$

Comme on retombe sur $\frac{5}{7}$ on déduit que $\frac{19}{7} = (10,\underline{101})_2$.

2. (a) i. la partie entière de 5,625 est $5 = (101)_2$. Par suite,

$$\begin{aligned} 0{,}625 \times 2 &= 1{,}25 && \to 1 \\ 0{,}25 \times 2 &= 0{,}5 && \to 0 \\ 0{,}5 \times 2 &= 1 && \to 1 \end{aligned}$$

Ainsi, $5{,}625 = 101{,}101$.

ii. Par suite, $5{,}625 = (1{,}01101)_2 \times 2^2$.

iii. $n = 2 = (10)_2 > 0$ donc son codage est `00000000010`

iv. Le code de $-5{,}625$ avec le pseudo-IEEE 754 sur 64 bits est alors :
`1000 0000 0010 0110 1000 0000 ... 0000`

(b) i. Le code `1000 0000 1101 1001 0000 ... 0000` est celui de $-(1{,}1001)_2 \times 2^{13}$ car l'exposant est codé par `000 0000 1101`, code d'un nombre positif car débutant par `0`, donc écriture binaire de l'exposant ; or $(1101)_2 = 2^3 + 2^2 + 2^0 = 13$.
Ensuite, $-(1{,}1001)_2 \times 2^{13} = -(11001000000000)_2$

ii. Finalement, le nombre codé est $-(2^{13} + 2^{12} + 2^9) = -12\ 800$.

Exercice 2 (5 points)

1. (a) Le contenu de la liste `li` à la fin de l'exécution du code est `[5,9,13]`. En effet, on ajoute les $2k+1$ quand k est paire, compris entre 2 inclus et 8 exclu, donc k valant 2, 4 et 6.

(b) Le code est : `li=[2*k+1 for k in range(2,8,2)]`

2. Le contenu de la liste `liste1` à la fin de l'exécution est `[1,2,3,4]`.
En effet, à chaque étape, on modifie l'élément d'indice `i` en lui ajoutant l'élément d'indice `i-1`, autrement dit l'élément précédent, en commençant par `i=1`. Ainsi, successivement, le contenu de `liste1` est : `[1,2,1,1]`, `[1,2,3,1]`, `[1,2,3,4]`.

3. Le code est :

```
def indice_val(liste,valeur):
    for i,x in enumerate(liste):
        if x == valeur:
            return i
    return None
```

LYCÉE MARIE CURIE **1NSI**
2022–2023

Devoir surveillé n°4 – NSI
Correction

Exercice 1 (5 points)

1. La partie entière de $\frac{11}{7}$ est $\frac{7}{7} = 1 = (1)_2$. La partie décimale est $\frac{4}{7}$. Par suite,

$$\begin{array}{ll} \frac{4}{7} \times 2 = \frac{8}{7} = \frac{7}{7} + \frac{1}{7} & \to 1 \\ \frac{1}{7} \times 2 = \frac{2}{7} & \to 0 \\ \frac{2}{7} \times 2 = \frac{4}{7} & \to 0 \end{array}$$

   Comme on retombe sur $\frac{4}{7}$ on déduit que $\frac{11}{7} = (1,\underline{100})_2$.

2. (a) i. la partie entière de 3,75 est $3 = (11)_2$. Par suite,

$$\begin{array}{ll} 0,75 \times 2 = 1,5 & \to 1 \\ 0,5 \times 2 = 1 & \to 1 \end{array}$$

   Ainsi, $3,75 = 11,11$.

   ii. Par suite, $3,75 = (1,111)_2 \times 2^1$.

   iii. $n = 1 = (1)_2 > 0$ donc son codage est `00000000001`

   iv. Le code de 3,75 avec le pseudo-IEEE 754 sur 64 bits est alors :
   `0000 0000 0001 1110 0000 ... 0000`

   (b) i. Le code `1000 0001 1010 0110 0000 ... 0000` est celui de $-(1,011)_2 \times 2^{26}$ car l'exposant est codé par `000 0001 1010`, code d'un nombre positif car débutant par `0`, donc écriture binaire de l'exposant ; or $(11010)_2 = 2^4 + 2^3 + 2^1 = 26$.
   Ensuite, $-(1,011)_2 \times 2^{26} = -(101100000000000000000000000)_2$

   ii. Finalement, le nombre codé est $-(2^{26} + 2^{24} + 2^{23}) = -92\ 274\ 688$.

Exercice 2 (5 points)

1. (a) Le contenu de la liste `lp` à la fin de l'exécution du code est `[2,6,10,14]`. En effet, on ajoute les $2k$ quand k est impair, compris entre 0 inclus et 8 exclu, donc k valant 1, 3, 5 et 7.

   (b) Le code est : `lp=[2*k for k in range(1,8,2)]`

2. Le contenu de la liste `liste1` à la fin de l'exécution est `[1,0,1,0]`.
   En effet, à chaque étape, on modifie l'élément d'indice `i` en lui soustrayant l'élément d'indice `i-1`, autrement dit l'élément précédent, en commençant par `i=1`. Ainsi, successivement, le contenu de `liste1` est : `[1,0,1,1]`, `[1,0,1,1]`, `[1,0,1,0]`.

3. Le code est :

```
def compte_val(liste,valeur):
    c=0
    for x in liste:
        if x == valeur:
            c+=1
    return c
```