Remarques

On remarque que

- de nombreux calculs impliquant des nombres à virgule ne sont pas correctements calculés par Python.

Remarques

On remarque que

- de nombreux calculs impliquant des nombres à virgule ne sont pas correctements calculés par Python.
  Un exemple classique est $0.1 + 0.2$ qui ne donne pas exactement 0.3, et par conséquent le test d'égalité $0.1 + 0.2 == 0.3$ renvoie `false`.

Remarques

On remarque que

- de nombreux calculs impliquant des nombres à virgule ne sont pas correctements calculés par Python.
  Un exemple classique est $0.1 + 0.2$ qui ne donne pas exactement 0.3, et par conséquent le test d'égalité $0.1 + 0.2 == 0.3$ renvoie `false`.
- lorsque des nombres trop grands sont en jeu on obtient un message d'erreur indiquant un dépassement de capacité *OverflowError*.

Remarques

On remarque que

- de nombreux calculs impliquant des nombres à virgule ne sont pas correctements calculés par Python.
  Un exemple classique est $0.1 + 0.2$ qui ne donne pas exactement 0.3, et par conséquent le test d'égalité $0.1 + 0.2 == 0.3$ renvoie `false`.
- lorsque des nombres trop grands sont en jeu on obtient un message d'erreur indiquant un dépassement de capacité *OverflowError*.

Le but du chapitre est de comprendre la représentation interne des nombres flottants qui conduit à ces résultats.

Ecriture dyadique

De la même façon que les chiffres après la virgule d'un nombre en écriture décimal utilisent les puissances de 10 négatives, par exemple :

Ecriture dyadique

De la même façon que les chiffres après la virgule d'un nombre en écriture décimal utilisent les puissances de 10 négatives, par exemple :

$\overline{14,05}^{10} =$

10^1	10^0	,	10^{-1}	10^{-2}
1	4	,	0	5

Ecriture dyadique

De la même façon que les chiffres après la virgule d'un nombre en écriture décimal utilisent les puissances de 10 négatives, par exemple :

$\overline{14,05}^{10} =$

10^1	10^0	,	10^{-1}	10^{-2}
1	4	,	0	5

En écriture binaire (ou dyadique) les chiffres après la virgule correspondent aux puissances négatives de 2 :

Ecriture dyadique

De la même façon que les chiffres après la virgule d'un nombre en écriture décimal utilisent les puissances de 10 négatives, par exemple :

$\overline{14,05}^{10} =$

10^1	10^0	,	10^{-1}	10^{-2}
1	4	,	0	5

En écriture binaire (ou dyadique) les chiffres après la virgule correspondent aux puissances négatives de 2 :

$\overline{10,01}^{2} =$

2^1	2^0	,	2^{-1}	2^{-2}
1	0	,	0	1

Ecriture dyadique

De la même façon que les chiffres après la virgule d'un nombre en écriture décimal utilisent les puissances de 10 négatives, par exemple :

$\overline{14,05}^{10} =$

10^1	10^0	,	10^{-1}	10^{-2}
1	4	,	0	5

En écriture binaire (ou dyadique) les chiffres après la virgule correspondent aux puissances négatives de 2 :

$\overline{10,01}^{2} =$

2^1	2^0	,	2^{-1}	2^{-2}
1	0	,	0	1

et donc $\overline{10,01}^{2} = \overline{2,25}^{10}$

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

Exemple

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.

Exemple

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

- $0,59375 \times 2 = 1,1875 \geq 1$ donc on ajoute 1 à l'écriture dyadique

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

- $0,59375 \times 2 = 1,1875 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,1875 \times 2 = 0,375 < 1$ donc on ajoute 0 à l'écriture dyadique

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

- $0,59375 \times 2 = 1,1875 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,1875 \times 2 = 0,375 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,375 \times 2 = 0,75 < 1$ donc on ajoute 0 à l'écriture dyadique

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

- $0,59375 \times 2 = 1,1875 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,1875 \times 2 = 0,375 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,375 \times 2 = 0,75 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,75 \times 2 = 1,5 \geq 1$ donc on ajoute 1 à l'écriture dyadique

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

- $0,59375 \times 2 = 1,1875 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,1875 \times 2 = 0,375 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,375 \times 2 = 0,75 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,75 \times 2 = 1,5 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,5 \times 2 = 1,0 \geq 1$ donc on ajoute 1 à l'écriture dyadique

Méthode : du décimal au dyadique

Pour traduire une partie décimale en écriture dyadique :

- Multiplier la partie décimale par 2. Si ce produit est supérieur ou égal à 1, ajouter 1 à l'écriture dyadique sinon ajouter 0.
- Recommencer avec la partie décimale de ce produit tant qu'elle est non nul.

Exemple

Par exemple si on veut écrire $\overline{0,59375}^{10}$ en binaire :

- $0,59375 \times 2 = 1,1875 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,1875 \times 2 = 0,375 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,375 \times 2 = 0,75 < 1$ donc on ajoute 0 à l'écriture dyadique
- $0,75 \times 2 = 1,5 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- $0,5 \times 2 = 1,0 \geq 1$ donc on ajoute 1 à l'écriture dyadique
- On s'arrête car la partie décimale du produit est 0 et $\overline{0,59375}^{10} = \overline{0,10011}^{2}$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^{2}$
   $\overline{1101,0111}^{2} = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^{2}$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique
   2. $0,2 \times 2 = 0,4 < 1$ donc on ajoute 0 à l'écriture dyadique

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique
   2. $0,2 \times 2 = 0,4 < 1$ donc on ajoute 0 à l'écriture dyadique
   3. $0,4 \times 2 = 0,8 < 1$ donc on ajoute 0 à l'écriture dyadique

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique
   2. $0,2 \times 2 = 0,4 < 1$ donc on ajoute 0 à l'écriture dyadique
   3. $0,4 \times 2 = 0,8 < 1$ donc on ajoute 0 à l'écriture dyadique
   4. $0,8 \times 2 = 1,6 \geq 1$ donc on ajoute 1 à l'écriture dyadique

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique
   2. $0,2 \times 2 = 0,4 < 1$ donc on ajoute 0 à l'écriture dyadique
   3. $0,4 \times 2 = 0,8 < 1$ donc on ajoute 0 à l'écriture dyadique
   4. $0,8 \times 2 = 1,6 \geq 1$ donc on ajoute 1 à l'écriture dyadique
   5. $0,6 \times 2 = 1,2 \geq 1$ donc on ajoute 1 à l'écriture dyadique

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique
   2. $0,2 \times 2 = 0,4 < 1$ donc on ajoute 0 à l'écriture dyadique
   3. $0,4 \times 2 = 0,8 < 1$ donc on ajoute 0 à l'écriture dyadique
   4. $0,8 \times 2 = 1,6 \geq 1$ donc on ajoute 1 à l'écriture dyadique
   5. $0,6 \times 2 = 1,2 \geq 1$ donc on ajoute 1 à l'écriture dyadique
   6. Le processus se poursuit indéfiniment car on est revenu à l'étape 2.

Exemples

1. Donner l'écriture décimale de $\overline{1101,0111}^2$
   $\overline{1101,0111}^2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} + 2^{-4} = \overline{13,4375}^{10}$
2. Donner l'écriture dyadique $3,5$
   $\overline{3,5}^{10} = \overline{11,1}^2$
3. Donner l'écriture dyadique $0,1$
   1. $0,1 \times 2 = 0,2 < 1$ donc on ajoute 0 à l'écriture dyadique
   2. $0,2 \times 2 = 0,4 < 1$ donc on ajoute 0 à l'écriture dyadique
   3. $0,4 \times 2 = 0,8 < 1$ donc on ajoute 0 à l'écriture dyadique
   4. $0,8 \times 2 = 1,6 \geq 1$ donc on ajoute 1 à l'écriture dyadique
   5. $0,6 \times 2 = 1,2 \geq 1$ donc on ajoute 1 à l'écriture dyadique
   6. Le processus se poursuit indéfiniment car on est revenu à l'étape 2.

   $\overline{0,1}^{10} = \overline{0,0001100110011\ldots}^2$

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

$$\boxed{\pm\, a \times 10^n}$$

Exemples

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

$$\boxed{\pm\, a \times 10^n}$$

- avec $a \in [1; 10[$, appelée mantisse (l'écriture décimal de a n'a qu'un seul chiffre non nul à gauche de la virgule)

Exemples

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

$$\boxed{\pm\, a \times 10^n}$$

- avec $a \in [1;10[$, appelée mantisse (l'écriture décimal de a n'a qu'un seul chiffre non nul à gauche de la virgule)
- et $n \in \mathbb{Z}$ appelée exposant.

Exemples

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

$$\boxed{\pm\, a \times 10^n}$$

- avec $a \in [1;10[$, appelée mantisse (l'écriture décimal de a n'a qu'un seul chiffre non nul à gauche de la virgule)
- et $n \in \mathbb{Z}$ appelée exposant.

Exemples

- $7200000000000 = 7,2 \times 10^{12}$.

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

$$\boxed{\pm\, a \times 10^n}$$

- avec $a \in [1;10[$, appelée mantisse (l'écriture décimal de a n'a qu'un seul chiffre non nul à gauche de la virgule)
- et $n \in \mathbb{Z}$ appelée exposant.

Exemples

- $7200000000000 = 7,2 \times 10^{12}$.
- $0,0000054 = 5,4 \times 10^{-6}$.

Ecriture scientifique

Ecrire un nombre en notation scientifique c'est l'écrire sous la forme

$$\boxed{\pm\, a \times 10^n}$$

- avec $a \in [1; 10[$, appelée mantisse (l'écriture décimal de a n'a qu'un seul chiffre non nul à gauche de la virgule)
- et $n \in \mathbb{Z}$ appelée exposant.

Exemples

- $7200000000000 = 7,2 \times 10^{12}$.
- $0,0000054 = 5,4 \times 10^{-6}$.
- 0 ne peut pas s'écrire en notation scientifique.

Virgule flottante

Les nombres non entiers en informatique, sont représentés en virgule flottante.
Cette représentation :

- se fonde sur l'écriture scientifique et utilise la base 2, c'est-à-dire l'écriture dyadique en utilisant une mantisse et un exposant de taille limitée.

Virgule flottante

Les nombres non entiers en informatique, sont représentés en virgule flottante.
Cette représentation :

- se fonde sur l'écriture scientifique et utilise la base 2, c'est-à-dire l'écriture dyadique en utilisant une mantisse et un exposant de taille limitée.
- La norme IEEE-754 définit deux formats codés respectivement sur 32 et 64 bits et stockés dans l'ordre signe/exposant/mantisse. Seul celui sur 64 bits est disponible en Python et correspond au type `float`

Virgule flottante

Les nombres non entiers en informatique, sont représentés en virgule flottante.
Cette représentation :

- se fonde sur l'écriture scientifique et utilise la base 2, c'est-à-dire l'écriture dyadique en utilisant une mantisse et un exposant de taille limitée.
- La norme IEEE-754 définit deux formats codés respectivement sur 32 et 64 bits et stockés dans l'ordre signe/exposant/mantisse. Seul celui sur 64 bits est disponible en Python et correspond au type `float`

	Signe	Exposant	Mantisse	Python
32 bits	1 bit	8 bits	23 bits	✗
64 bits	1 bit	11 bits	52 bits	`float`

Virgule flottante

Les nombres non entiers en informatique, sont représentés en virgule flottante.
Cette représentation :

- se fonde sur l'écriture scientifique et utilise la base 2, c'est-à-dire l'écriture dyadique en utilisant une mantisse et un exposant de taille limitée.
- La norme IEEE-754 définit deux formats codés respectivement sur 32 et 64 bits et stockés dans l'ordre signe/exposant/mantisse. Seul celui sur 64 bits est disponible en Python et correspond au type `float`

	Signe	Exposant	Mantisse	Python
32 bits	1 bit	8 bits	23 bits	✗
64 bits	1 bit	11 bits	52 bits	`float`

- L'exposant est décalé de façon à toujours être stocké sous la forme d'un entier positif. Ce décalage est de $127(=2^8-1)$ pour le format 32 bits et de $1023(=2^{11}-1)$ pour le format 64 bits.

Virgule flottante

Les nombres non entiers en informatique, sont représentés en virgule flottante.
Cette représentation :

- se fonde sur l'écriture scientifique et utilise la base 2, c'est-à-dire l'écriture dyadique en utilisant une mantisse et un exposant de taille limitée.
- La norme IEEE-754 définit deux formats codés respectivement sur 32 et 64 bits et stockés dans l'ordre signe/exposant/mantisse. Seul celui sur 64 bits est disponible en Python et correspond au type `float`

	Signe	Exposant	Mantisse	Python
32 bits	1 bit	8 bits	23 bits	✗
64 bits	1 bit	11 bits	52 bits	`float`

- L'exposant est décalé de façon à toujours être stocké sous la forme d'un entier positif. Ce décalage est de $127 (= 2^8 - 1)$ pour le format 32 bits et de $1023 (= 2^{11} - 1)$ pour le format 64 bits.
- Certaines valeurs spéciales de l'exposant et de la mantisse servent à représenter des valeurs particulières (infinis, zéros, NaN).

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

1. Le nombre est négatif, donc le bit de signe est $\boxed{1}$.

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

1. Le nombre est négatif, donc le bit de signe est $\boxed{1}$.
2. $\overline{168,75}^{10} = \overline{10101000,11}^{2}$

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

1. Le nombre est négatif, donc le bit de signe est $\boxed{1}$.
2. $\overline{168,75}^{10} = \overline{10101000,11}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{168,75}^{10} = \overline{1,010100011}^{2} \times 2^7$

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

1. Le nombre est négatif, donc le bit de signe est $\boxed{1}$.
2. $\overline{168,75}^{10} = \overline{10101000,11}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{168,75}^{10} = \overline{1,010100011}^{2} \times 2^7$
4. L'exposant est donc 7, et avec le décalage il est stocké sous la forme 7+127 = 134. c'est-à-dire $\boxed{10\,000\,110}$ en base 2.

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

1. Le nombre est négatif, donc le bit de signe est $\boxed{1}$.
2. $\overline{168,75}^{10} = \overline{10101000,11}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{168,75}^{10} = \overline{1,010100011}^{2} \times 2^7$
4. L'exposant est donc 7, et avec le décalage il est stocké sous la forme 7+127 = 134. c'est-à-dire $\boxed{10\,000\,110}$ en base 2.
5. On complète la mantisse par des zéros de façon à avoir 23 bits et le 1 initial n'est pas stocké afin d'économiser un bit. La mantisse est donc
   $\boxed{01\,010\,001\,100\,000\,000\,000\,000}$

Exemple 1

Donner le représentation sur 32 bits du nombre $-168,75$

1. Le nombre est négatif, donc le bit de signe est $\boxed{1}$.
2. $\overline{168,75}^{10} = \overline{10101000,11}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{168,75}^{10} = \overline{1,010100011}^{2} \times 2^{7}$
4. L'exposant est donc 7, et avec le décalage il est stocké sous la forme $7+127 = 134$. c'est-à-dire $\boxed{10\,000\,110}$ en base 2.
5. On complète la mantisse par des zéros de façon à avoir 23 bits et le 1 initial n'est pas stocké afin d'économiser un bit. La mantisse est donc $\boxed{01\,010\,001\,100\,000\,000\,000\,000}$

Le nombre $-168,75$ est donc stocké sous la forme :
$\boxed{1}\boxed{10\,000\,110}\boxed{01\,010\,001\,100\,000\,000\,000\,000}$

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

1. Le nombre est positif, donc le bit de signe est $\boxed{0}$.

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

1. Le nombre est positif, donc le bit de signe est $\boxed{0}$.
2. $\overline{0,1}^{10} = \overline{0,000110011001100\ldots}^{2}$

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

1. Le nombre est positif, donc le bit de signe est $\boxed{0}$.
2. $\overline{0,1}^{10} = \overline{0,000110011001100\ldots}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{0,1}^{10} = \overline{1,10011001100\ldots}^{2} \times 2^{-4}$

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

1. Le nombre est positif, donc le bit de signe est $\boxed{0}$.
2. $\overline{0,1}^{10} = \overline{0,000110011001100\dots}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{0,1}^{10} = \overline{1,10011001100\dots}^{2} \times 2^{-4}$
4. L'exposant est donc -4, et avec le décalage il est stocké sous la forme $-4 + 127 = 123$. c'est-à-dire $\boxed{01\,111\,011}$ en base 2.

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

1. Le nombre est positif, donc le bit de signe est $\boxed{0}$.
2. $\overline{0,1}^{10} = \overline{0,000110011001100\ldots}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{0,1}^{10} = \overline{1,10011001100\ldots}^{2} \times 2^{-4}$
4. L'exposant est donc -4, et avec le décalage il est stocké sous la forme $-4+127 = 123$. c'est-à-dire $\boxed{01\,111\,011}$ en base 2.
5. La mantisse est infinie, on la limite au 23 premiers bits (c'est un arrondi et non une troncature) le 1 initial n'est pas stocké afin d'économiser un bit. La mantisse est donc $\boxed{10\,011\,001\,100\,110\,011\,001\,101}$

Exemple 2

Donner le représentation sur 32 bits du nombre $0,1$

1. Le nombre est positif, donc le bit de signe est $\boxed{0}$.
2. $\overline{0,1}^{10} = \overline{0,000110011001100\dots}^{2}$
3. La mantisse est décalée de façon à n'avoir qu'un chiffre non nul avant la virgule :
   $\overline{0,1}^{10} = \overline{1,10011001100\dots}^{2} \times 2^{-4}$
4. L'exposant est donc -4, et avec le décalage il est stocké sous la forme $-4+127=123$. c'est-à-dire $\boxed{01\,111\,011}$ en base 2.
5. La mantisse est infinie, on la limite au 23 premiers bits (c'est un arrondi et non une troncature) le 1 initial n'est pas stocké afin d'économiser un bit. La mantisse est donc $\boxed{10\,011\,001\,100\,110\,011\,001\,101}$

Le nombre $0,1$ est donc stocké sous la forme :
$\boxed{0}\boxed{01\,111\,011}\boxed{10\,011\,001\,100\,110\,011\,001\,101}$

Exemple 3

Quel nombre est stocké sous la forme :

$\boxed{0}\,\boxed{10\,000\,100}\,\boxed{01\,010\,101\,000\,000\,000\,000\,000}$

1. Le bit de signe est 0, le nombre est positif

Exemple 3

Quel nombre est stocké sous la forme :

$\boxed{0}\,\boxed{10\,000\,100}\,\boxed{01\,010\,101\,000\,000\,000\,000\,000}$

1. Le bit de signe est 0, le nombre est positif
2. L'exposant est $\overline{10000100}^{2} = \overline{132}^{10}$, c'est-à-dire 5 en soustrayant le décalage de 127.

Exemple 3

Quel nombre est stocké sous la forme :

0	10 000 100	01 010 101 000 000 000 000 000

1. Le bit de signe est 0, le nombre est positif
2. L'exposant est $\overline{10000100}^{2} = \overline{132}^{10}$, c'est-à-dire 5 en soustrayant le décalage de 127.
3. Le 1 initial de la mantisse n'est pas stocké et donc elle est en réalité de $\overline{1,01010101000000000000000}^{2} = \overline{1,33203125}^{10}$

Exemple 3

Quel nombre est stocké sous la forme :

$\boxed{0}\boxed{10\,000\,100}\boxed{01\,010\,101\,000\,000\,000\,000\,000}$

1. Le bit de signe est 0, le nombre est positif
2. L'exposant est $\overline{10000100}^{2} = \overline{132}^{10}$, c'est-à-dire 5 en soustrayant le décalage de 127.
3. Le 1 initial de la mantisse n'est pas stocké et donc elle est en réalité de $\overline{1,01010101000000000000000}^{2} = \overline{1,33203125}^{10}$
4. Ce nombre est donc $1,33203125 \times 2^5 = 42,625$.

Attention !

Cette représentation approximative des nombres réels induit des conséquences importantes :

Attention !

Cette représentation approximative des nombres réels induit des conséquences importantes :

- Les tests d'égalité entre flottants ne sont pas pertinents. On doit les éviter ou les effectuer à un ε près.

Attention !

Cette représentation approximative des nombres réels induit des conséquences importantes :

- Les tests d'égalité entre flottants ne sont pas pertinents. On doit les éviter ou les effectuer à un ε près.
  A titre d'exemple le test `0.1 + 0.2 == 0.3` renvoie faux

Attention !

Cette représentation approximative des nombres réels induit des conséquences importantes :

- Les tests d'égalité entre flottants ne sont pas pertinents. On doit les éviter ou les effectuer à un ε près.
  A titre d'exemple le test `0.1 + 0.2 == 0.3` renvoie faux
- Les valeurs calculées par un programme peuvent être très éloignés des valeurs théoriques d'un algorithme.

Attention !

Cette représentation approximative des nombres réels induit des conséquences importantes :

- Les tests d'égalité entre flottants ne sont pas pertinents. On doit les éviter ou les effectuer à un ε près.
  A titre d'exemple le test `0.1 + 0.2 == 0.3` renvoie faux
- Les valeurs calculées par un programme peuvent être très éloignés des valeurs théoriques d'un algorithme.
  Des exemples seront vus en TP.