

Représentation des données — Binaire, hexadécimal

NSI · Première
Contrôle n°1 · 60 min · sur 20

Exercice 1 — Compter en base 2 et en base 16

5 pts 14 min

Convertir entre bases

Réponds à chaque question.

LA RÈGLE DU COURS

Deux bases se pratiquent en informatique, et l'une abrège l'autre sans en trahir la structure.

1. Que vaut un bit, et combien de bits compte un octet ? (1 pt)

Un bit vaut 0 ou 1 ; un octet compte 8 bits.

→ Chaque position vaut une puissance de 2 croissante de droite à gauche.

2. Convertis 1011 en base 2 vers le décimal. (1 pt)

11.

→ On additionne 8, 2 et 1, les positions dont le bit vaut 1.

3. Combien de valeurs différentes un octet peut-il représenter ? (1 pt)

256 valeurs, de 0 à 255 pour des entiers non signés.

→ Chaque bit prend deux valeurs, et les huit bits sont indépendants.

4. Quels chiffres l'hexadécimal emploie-t-il, et à quoi correspond un chiffre ? (1 pt)

Les chiffres 0 à 9 puis les lettres A à F pour 10 à 15 ; un chiffre correspond exactement à quatre bits.

→ Un octet s'écrit donc avec deux chiffres hexadécimaux.

5. Que vaut FF en hexadécimal, en décimal ? (1 pt)

255.

→ C'est 15 fois 16 plus 15, soit la plus grande valeur tenant sur un octet.

Exercice 2 — Les entiers négatifs

4 pts 11 min

Analyser le complément à 2

Réponds à chaque question.

LA RÈGLE DU COURS

Une représentation code les nombres négatifs sans signe séparé, et son intérêt est décisif pour le matériel.

1. Que vaut le bit de poids fort en complément à 2 sur 8 bits ? (1 pt)

-128.

→ Les sept autres bits gardent leurs valeurs positives.

2. Quel est l'intérêt décisif de cette représentation ? (1 pt)

L'addition fonctionne sans cas particulier : le même circuit additionne des nombres positifs et négatifs.

→ C'est ce qui justifie ce choix plutôt qu'un bit de signe séparé.

3. Quelle plage d'entiers représente-t-on ainsi sur 8 bits ? (1 pt)

De -128 à 127.

→ Le total fait bien 256 valeurs, comme pour les entiers non signés.

4. Que se passe-t-il si l'on ajoute 1 à la plus grande valeur représentable ? (1 pt)

On obtient la plus petite : le compteur repart à l'autre extrémité.

→ Le résultat est parfaitement déterministe et parfaitement faux.

Exercice 3 — Coder les caractères

4 pts 11 min

Analyser le codage des caractères

Réponds à chaque question.

LA RÈGLE DU COURS

Une table historique ne suffit plus, et son successeur a une propriété qui a des conséquences pratiques constantes.

1. Sur combien de bits la table ASCII code-t-elle, et combien de caractères couvre-t-elle ? (1 pt)

Sur 7 bits, soit 128 caractères.

→ Elle ne couvre que l'alphabet latin non accentué.

2. Comment UTF-8 résout-il le problème des autres écritures ? (1 pt)

Par un codage de longueur variable : un, deux, trois ou quatre octets selon le caractère.

→ Les caractères ASCII y gardent leur code sur un octet, ce qui assure la compatibilité.

3. Pourquoi un texte accentué occupe-t-il plus d'octets que de caractères ? (1 pt)

Parce que les caractères hors ASCII occupent plusieurs octets chacun.

→ La longueur en caractères et la taille en octets ne coïncident donc pas.

4. Quelle erreur pratique cette confusion produit-elle ? (1 pt)

Des chaînes tronquées au milieu d'un caractère.

→ C'est l'origine des caractères illisibles qu'on voit encore apparaître.

Exercice 4 — Les nombres à virgule

4 pts 13 min

Analyser les flottants

Réponds à chaque question.

LA RÈGLE DU COURS

Une norme code les nombres à virgule sur un nombre fini de bits, et certaines valeurs simples n'y tiennent pas.

1. Selon quelle norme un flottant est-il codé, et de quoi se compose-t-il ? (1 pt)

Selon la norme IEEE 754 : un signe, un exposant et une mantisse.

→ Le tout tient sur un nombre fini de bits.

2. Pourquoi 0,1 n'est-il pas stocké exactement ? (1 pt)

Parce qu'il n'a pas d'écriture finie en base 2.

→ C'est le même phénomène qu'un tiers, qui n'en a pas en base 10.

3. Pourquoi une égalité stricte entre flottants peut-elle échouer sans faute de programmation ? (1 pt)

Parce qu'elle porte sur des approximations, et que rien ne garantit que deux chemins de calcul aboutissent au même arrondi.

→ Aucun des nombres comparés n'est stocké exactement.

4. Comment faut-il comparer deux flottants ? (1 pt)

En vérifiant que l'écart absolu est inférieur à un seuil choisi.

→ On compare un écart à une tolérance, jamais deux valeurs.

Repérer une erreur et la corriger

Chaque affirmation est fausse. Écris ce qu'il fallait dire.

LA RÈGLE DU COURS

Trois fautes reviennent. Une imprécision attribuée à la machine, deux grandeurs confondues, et une plage mal bornée.

1. « Si 0,1 plus 0,2 ne fait pas 0,3, c'est que la machine manque de précision. » (1 pt)

C'est une conséquence directe de la base 2 : 0,1 n'a pas d'écriture finie en base 2.

→ Le flottant n'en stocke qu'une approximation.

2. « Cette chaîne compte 10 caractères : elle occupe donc 10 octets. » (1 pt)

En UTF-8, un caractère accentué occupe plusieurs octets.

→ La longueur en caractères et la taille en octets ne coïncident pas.

3. « Pour convertir du binaire vers l'hexadécimal, il faut d'abord passer par le décimal. » (1 pt)

On regroupe les bits par quatre : aucune division n'est nécessaire.

→ Chaque groupe de quatre bits donne directement un chiffre hexadécimal.

Barème — ce que le correcteur regarde

Les lignes chiffrées couvrent les 5 exercices et totalisent 20 points.

EXERCICE	ATTENDU	POINTS	POINT PERDU SI
Ex.1	Binaire et hexadécimal	5 points — 1 par question	Une puissance de 2 comptée dans le mauvais sens.
Ex.2	Les entiers négatifs	4 points — 1 par question	Un bit de signe séparé inventé.
Ex.3	Coder les caractères	4 points — 1 par question	Longueur en caractères et taille en octets confondues.
Ex.4	Les nombres à virgule	4 points — 1 par question	Une égalité stricte appliquée à des flottants.
Ex.5	L'erreur repérée, et la réponse juste donnée	3 points — 1 par ligne	Une correction qui attribue le problème à la machine.

ET MAINTENANT ?

16 à 20

Le chapitre est acquis. Entraîne-toi maintenant à convertir en cinq temps : identifier les deux bases, additionner les puissances vers le décimal, diviser et lire les restes de bas en haut depuis le décimal, regrouper par quatre entre binaire et hexadécimal, et vérifier l'ordre de grandeur.

10 à 15

Refais l'exercice 1 en écrivant les puissances de 2 au-dessus de chaque bit avant de convertir.

moins de 10

Reprends la fiche de cours et retiens pourquoi 0,1 plus 0,2 ne fait pas 0,3 : ce nombre n'a pas d'écriture finie en base 2.