



SPÉ NSI · 1ÈRE

✓ EXIGIBLE AU BAC

PRIORITÉ P1

Représentation binaire des entiers

conversion et dépassement de capacité

À LA FIN DE CETTE FICHE, TU SAURAS...

- ✓ Convertir un entier naturel de la base 10 vers la base 2, et inversement.
- ✓ Écrire un entier relatif (positif ou négatif) en complément à deux sur un nombre de bits donné.
- ✓ Reconnaître un dépassement de capacité lors d'un calcul sur des entiers codés sur un nombre de bits fixé.

TON PARCOURS DE RÉVISION — coche chaque partie une fois révisée !

- | | |
|--|---|
| ☐ 1 · Les mots à connaître | ☐ 8 · La carte mentale |
| ☐ 2 · Les règles essentielles | ☐ 9 · Trouve l'erreur |
| ☐ 3 · Méthode A | ☐ 10 · Exercices type Bac |
| ☐ 4 · Méthode B | ☐ 11 · QCM chrono (5 min) |
| ☐ 5 · Exemple type Bac | ☐ 12 · Ta carte « 30 secondes » |
| ☐ 6 · Les pièges à éviter | ☐ ∞ Corrections (pour les parents) |
| ☐ 7 · Tips pour t'en souvenir | |

LE COIN DES PARENTS — pas besoin de connaître le cours, tout est justifié.

- Consultez avec votre enfant le calendrier des échéances (devoirs, DS, épreuves de spécialité, Grand Oral) et vérifiez qu'aucune date n'est oubliée.
- Aidez-le à répartir les révisions de ce chapitre sur plusieurs jours (planning de répétition espacée J+1/J+3/J+7/J+14), sans vous prononcer sur le contenu du programme.
- Observez les signaux d'alerte (fatigue persistante, stress, décrochage) et parlez-en avec lui, sans vous substituer aux professionnels du lycée.

1 Les mots à connaître *surligne ceux que tu confonds !*

Bit — la plus petite unité d'information en informatique, qui ne prend que deux valeurs possibles : 0 ou 1.

Octet — un groupe de 8 bits, unité courante pour mesurer une quantité de mémoire.

Base 2 (binaire) — un système de numération qui n'utilise que deux chiffres, 0 et 1, où chaque position vaut une puissance de 2.

Base 10 (décimale) — le système de numération usuel, qui utilise dix chiffres (0 à 9), où chaque position vaut une puissance de 10.

Poids d'un bit — la valeur (une puissance de 2) associée à la position d'un bit dans une écriture binaire.

Entier naturel — un nombre entier positif ou nul (0, 1, 2, 3...), sans signe à représenter.

Entier relatif — un nombre entier qui peut être positif, négatif ou nul, et dont le codage doit donc représenter le signe.

Complément à deux — la méthode standard de codage des entiers relatifs en binaire, où le bit de poids fort indique le signe et où l'opposé d'un nombre s'obtient en inversant tous les bits puis en ajoutant 1.

Bit de poids fort — le bit le plus à gauche d'une écriture binaire, celui qui a le plus grand poids ; en complément à deux, il indique le signe du nombre.

Dépassement de capacité — la situation où le résultat d'un calcul ne peut pas être représenté avec le nombre de bits disponibles, ce qui produit un résultat incorrect.

2 Les règles essentielles *chaque règle avec son exemple*

- 1 **Un entier naturel s'écrit en base 2 comme une somme de puissances de 2 : chaque bit indique si la puissance de 2 correspondante (son poids) est utilisée (1) ou non (0).**

exemple → 1101 en base 2 correspond à $1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$ en base 10.

- 2 **Pour convertir un entier naturel de la base 10 vers la base 2, on effectue des divisions successives par 2 et on lit les restes obtenus, du dernier au premier.**

exemple → $13 = 2 \times 6 + 1$ (quotient 6, reste 1) ; $6 = 2 \times 3 + 0$ (quotient 3, reste 0) ; $3 = 2 \times 1 + 1$ (quotient 1, reste 1) ; $1 = 2 \times 0 + 1$ (quotient 0, reste 1). En lisant les restes du dernier au premier : 1101.

- 3 **Sur n bits, le complément à deux code les entiers relatifs de $-2^{(n-1)}$ à $2^{(n-1)} - 1$; le bit de poids fort vaut 0 pour un nombre positif ou nul, 1 pour un nombre négatif.**

exemple → Sur 4 bits, les entiers relatifs codables vont de -8 à 7 ; 0101 (bit de poids fort = 0) code un nombre positif, 1101 (bit de poids fort = 1) code un nombre négatif.

- 4 **Pour obtenir l'opposé d'un nombre en complément à deux, on inverse tous les bits (0 devient 1, 1 devient 0), puis on ajoute 1 au résultat.**

exemple → Sur 4 bits, 3 s'écrit 0011. On inverse les bits : 1100. On ajoute 1 : 1101, qui code donc -3.

- 5 **Un dépassement de capacité se produit quand le résultat d'une addition dépasse la plus grande valeur représentable sur le nombre de bits disponibles : le résultat calculé devient alors incorrect.**

exemple → Sur 4 bits non signés (de 0 à 15), $15 + 1$ ne peut pas se coder sur 4 bits : le calcul donne un résultat incorrect au lieu de 16.

- 6 **Pour lire la valeur codée par un octet non signé (8 bits), on additionne les puissances de 2 des bits à 1, de 2^7 (poids fort, à gauche) à 2^0 (à droite).**

exemple → L'octet 00010110 vaut $0 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 16 + 4 + 2 = 22$.

3

Méthode A — Convertir un entier naturel de la base 10 vers la base 2 (divisions successives)

Exemple suivi : convertir 22 en base 2.

Étape 1 Diviser 22 par 2 : $22 = 2 \times 11 + 0$ (quotient 11, reste 0).

Étape 2 Diviser le quotient obtenu par 2, et répéter jusqu'à obtenir un quotient nul : $11 = 2 \times 5 + 1$ (quotient 5, reste 1) ; $5 = 2 \times 2 + 1$ (quotient 2, reste 1) ; $2 = 2 \times 1 + 0$ (quotient 1, reste 0) ; $1 = 2 \times 0 + 1$ (quotient 0, reste 1).

Étape 3 Lire les restes obtenus du dernier au premier pour former l'écriture binaire : 10110.

Étape 4 Vérifier en reconvertissant : $1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 16 + 4 + 2 = 22$.

4

Méthode B — Coder un entier négatif en complément à deux sur n bits

Exemple suivi : coder -5 sur 4 bits.

Étape 1 Écrire la valeur absolue du nombre en binaire sur n bits : 5 s'écrit 0101 sur 4 bits.

Étape 2 Inverser tous les bits (0 devient 1, 1 devient 0) : 0101 devient 1010.

Étape 3 Ajouter 1 au résultat obtenu : 1010, plus 1, donne 1011.

Étape 4 Vérifier que le bit de poids fort vaut bien 1 (nombre négatif) : c'est le cas, 1011 code donc -5 sur 4 bits.

5

Exemple type Bac — rédigé pas à pas

Sur 4 bits, quel est le résultat de l'addition $0110 + 0011$ en binaire ? Vérifie ta réponse en convertissant chaque terme et le résultat en base 10, et indique si un dépassement de capacité se produit (on considère des entiers naturels non signés, codables de 0 à 15 sur 4 bits).

Étape 1 — Convertir chaque terme en base 10 pour vérifier : 0110 vaut $0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 6$; 0011 vaut $0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 3$.

Étape 2 — Additionner en base 10 pour connaître le résultat attendu : $6 + 3 = 9$.

Étape 3 — Vérifier que 9 est bien représentable sur 4 bits non signés (de 0 à 15) : 9 est inférieur à 15, donc pas de dépassement de capacité.

Étape 4 — Écrire 9 en base 2 par divisions successives : $9 = 2 \times 4 + 1$ (quotient 4, reste 1) ; $4 = 2 \times 2 + 0$ (quotient 2, reste 0) ; $2 = 2 \times 1 + 0$ (quotient 1, reste 0) ; $1 = 2 \times 0 + 1$ (quotient 0, reste 1). En lisant les restes du dernier au premier : 1001.

Conclusion : 0110 additionné à 0011 donne 1001 en binaire, soit 9 en base 10 : aucun dépassement de capacité, 9 est représentable sur 4 bits non signés (de 0 à 15).

6 Les pièges à éviter *les erreurs classiques en DS...*

X Confondre le bit de poids fort (le plus à gauche, la plus grande valeur) et le bit de poids faible (le plus à droite, la plus petite valeur).

X Oublier d'ajouter 1 après avoir inversé les bits lors du codage d'un nombre négatif en complément à deux : inverser les bits seul ne donne pas l'opposé correct.

X Croire qu'un dépassement de capacité provoque un message d'erreur automatique : il produit silencieusement un résultat incorrect, sans signal, si rien ne le détecte.

X Confondre la base 2 (binaire, deux chiffres 0 et 1) avec la base 10 (décimale, dix chiffres 0 à 9) lors d'une conversion.

7 Tips — pour t'en souvenir

Pour convertir décimal → binaire : divise par 2 en boucle, note les restes, puis lis-les du DERNIER au PREMIER.

Complément à deux d'un nombre négatif = inverser TOUS les bits, PUIS ajouter 1 — jamais l'un sans l'autre.

Le bit de poids fort (le plus à gauche) donne le signe en complément à deux : 0 = positif ou nul, 1 = négatif.

8 La carte mentale *dessiner, c'est mémoriser !*

Carte mentale à dessiner : « REPRÉSENTATION BINAIRE » au centre, 3 branches colorées : CONVERSION (divisions successives par 2, restes du dernier au premier), COMPLÉMENT À DEUX (inverser les bits puis ajouter 1, bit de poids fort = signe), DÉPASSEMENT DE CAPACITÉ (résultat incorrect si trop grand pour le nombre de bits). Un exemple chiffré pour chaque branche. Redessine-la de mémoire après avoir lu la fiche.



ta carte mentale, de mémoire...

Pour coder -5 sur 4 bits en complément à deux, un élève écrit : « 5 s'écrit 0101 en binaire, j'inverse les bits pour obtenir 1010, donc -5 s'écrit 1010. »

→ *cherche l'erreur avant de lire la réponse !*

L'erreur : L'élève a oublié la dernière étape du complément à deux : après avoir inversé les bits, il faut encore ajouter 1 au résultat obtenu. 1010 est le résultat de l'inversion des bits, mais pas encore le complément à deux complet : il faut encore ajouter 1, ce qui donne 1011. C'est 1011, et non 1010, qui code -5 sur 4 bits. Réflexe : complément à deux = inverser les bits, PUIS ajouter 1 — les deux étapes sont indispensables.

10 Exercices type Bac *corrections en fin de fiche ✂*

Exercice 1. Convertis l'entier naturel 18 de la base 10 vers la base 2, par divisions successives.

Exercice 2. Code l'entier -6 en complément à deux sur 4 bits.

Une seule bonne réponse par question, sans calculatrice — entraînement épreuve d'automatismes du Bac 2026.

1. Que vaut 1101 en base 2, converti en base 10 ?

- ☐ A 11
☐ B 13
☐ C 14

2. Combien de valeurs différentes peut-on coder avec un octet (8 bits) ?

- ☐ A 8
☐ B 16
☐ C 256

3. Sur 4 bits, quelle est la plus grande valeur représentable pour un entier naturel non signé (sans signe) ?

- ☐ A 15
☐ B 16
☐ C 8

4. Sur 4 bits, en complément à deux, quel est l'intervalle des entiers relatifs codables ?

- ☐ A de -15 à 15
☐ B de -7 à 8
☐ C de -8 à 7

5. Pour coder l'opposé d'un nombre en complément à deux, il faut :

- ☐ A inverser tous les bits, sans rien ajouter
- ☐ B inverser tous les bits, puis ajouter 1
- ☐ C ajouter 1, sans inverser les bits

6. Dans une écriture binaire en complément à deux, à quoi sert le bit de poids fort (le plus à gauche) ?

- ☐ A il indique le signe du nombre (0 = positif ou nul, 1 = négatif)
- ☐ B il indique toujours la valeur 1
- ☐ C il n'a aucune signification particulière

7. Sur 4 bits non signés (de 0 à 15), que se passe-t-il si on calcule $15 + 1$?

- ☐ A le résultat est 16, correctement représenté
- ☐ B un dépassement de capacité se produit, le résultat calculé devient incorrect
- ☐ C le calcul est refusé et rien ne s'affiche

8. Que vaut l'octet 00010110 converti en base 10 ?

- ☐ A 16
- ☐ B 22
- ☐ C 38

9. Pour convertir un entier naturel de la base 10 vers la base 2, quelle méthode utilise-t-on ?

- ☐ A des divisions successives par 2, en lisant les restes du dernier au premier
- ☐ B des divisions successives par 10, en lisant les restes du premier au dernier
- ☐ C une simple multiplication par 2

10. Sur 4 bits, à quel entier relatif correspond l'écriture binaire 1011 en complément à deux ?

- ☐ A 11
- ☐ B -3
- ☐ C -5



12

En 30 secondes — ta carte de dernière révision

découpe-la et garde-la dans ta trousse — révision de veille de contrôle !

- Binaire = base 2, deux chiffres (0 et 1), chaque position vaut une puissance de 2.
- Décimal → binaire : divisions successives par 2, restes lus du dernier au premier.
- Complément à deux : inverser les bits puis ajouter 1 pour coder l'opposé d'un nombre ; le bit de poids fort indique le signe.

PLAN DE RÉVISION ESPACÉE — relis la carte et refais le QCM : ☐ J+1 ☐ J+3 ☐ J+7 ☐ J+14



QCM en ligne

VA PLUS LOIN — TESTE-TOI SUR CETTE NOTION

Scanne le QR code (ou clique dessus) : le QCM chronométré de cette notion, note immédiate, chaque erreur expliquée. Gratuit, sans inscription.

Pour t'entraîner — QCM en nombre, évaluations, explications détaillées, analyse de tes points faibles — l'abonnement Réussito (sans engagement) : via le QR code ou sur reussito.fr/abo

Débloque toutes les fiches de 1ère : via le QR code ou sur reussito.fr/premiere



L'élève travaille sans les solutions sous les yeux ; vous corrigez ensemble avec ces pages. Chaque réponse est justifiée : pas besoin de connaître le cours.

Correction des exercices

Exercice 1. $18 = 2 \times 9 + 0$ (quotient 9, reste 0) ; $9 = 2 \times 4 + 1$ (quotient 4, reste 1) ; $4 = 2 \times 2 + 0$ (quotient 2, reste 0) ; $2 = 2 \times 1 + 0$ (quotient 1, reste 0) ; $1 = 2 \times 0 + 1$ (quotient 0, reste 1). En lisant les restes du dernier au premier : 10010. Vérification : $1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 16 + 2 = 18$.

Exercice 2. 6 s'écrit 0110 sur 4 bits. On inverse les bits : 1001. On ajoute 1, ce qui donne 1010. Donc -6 s'écrit 1010 sur 4 bits.

Réponses du QCM — expliquées une par une

- 1 → B** 1101 vaut $1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 8 + 4 + 1 = 13$. Piège : 11 correspondrait à une erreur de calcul des poids, par exemple oublier le bit de poids 4 ; 14 correspondrait à compter à tort un bit à 1 pour le bit de poids 2, qui vaut pourtant 0 dans 1101.
- 2 → C** Un octet compte 8 bits, chacun valant 0 ou 1 : il permet de coder 2 multiplié par lui-même 8 fois, soit 256 valeurs différentes. Piège : 8 correspondrait au nombre de bits lui-même, pas au nombre de valeurs codables ; 16 correspondrait au nombre de valeurs codables sur 4 bits, pas sur 8.
- 3 → A** Sur n bits non signés, la plus grande valeur représentable est 2 puissance n, moins 1 : sur 4 bits, 2 puissance 4 vaut 16, donc 16 moins 1 égale 15. Piège : 16 correspondrait à 2 puissance 4, une valeur non représentable sur 4 bits, il faudrait un 5e bit ; 8 correspondrait à 2 puissance 3, la plus grande puissance de 2 utilisable, mais pas la plus grande valeur codable.
- 4 → C** Sur n bits en complément à deux, l'intervalle codable va de $-2^{(n-1)}$ à $2^{(n-1)} - 1$: sur 4 bits, de -8 à 7. Piège : de -7 à 8 inverse à tort les deux bornes, la borne négative va plus loin que la positive, pas l'inverse ; de -15 à 15 correspondrait à un intervalle symétrique qui ne respecte pas la formule du complément à deux.
- 5 → B** Le complément à deux exige les deux étapes dans l'ordre : inverser tous les bits, PUIS ajouter 1 au résultat obtenu. Piège : inverser les bits seul ne donne que le complément à un, pas l'opposé correct en complément à deux ; ajouter 1 sans inverser les bits ne produit pas non plus l'opposé du nombre.
- 6 → A** En complément à deux, le bit de poids fort indique le signe du nombre codé : 0 pour un nombre positif ou nul, 1 pour un nombre négatif. Piège : le bit de poids fort peut valoir 0 ou 1 selon le signe du nombre, pas toujours 1 ; il a au contraire une signification précise, celle du signe, ce n'est pas un bit sans rôle.
- 7 → B** 16 dépasse la plus grande valeur représentable sur 4 bits non signés (15) : un dépassement de capacité se produit, et le résultat calculé devient incorrect. Piège : 16 ne peut justement pas être représenté sur 4 bits, ce n'est donc pas un résultat correct ; un dépassement de capacité ne bloque pas le calcul par un refus, il produit silencieusement un résultat faux.
- 8 → B** 00010110 vaut $1 \times 16 + 1 \times 4 + 1 \times 2 = 16 + 4 + 2 = 22$, les autres bits valant 0. Piège : 16 ne compte que le bit de poids 16, en oubliant les bits de poids 4 et 2 également à 1 ; 38 correspondrait à une erreur d'addition des poids des bits à 1.
- 9 → A** La conversion décimal vers binaire s'obtient par divisions successives par 2 : on lit les restes obtenus, du dernier au premier. Piège : diviser par 10 correspondrait à rester en base 10, ce n'est pas la conversion vers la base 2 ; une simple multiplication par 2 ne donne pas l'écriture binaire d'un nombre.

10 → C

Le bit de poids fort vaut 1 : le nombre est négatif. Sa valeur est $-8 + (0 \times 4 + 1 \times 2 + 1 \times 1) = -8 + 3 = -5$. Piège : -3 correspondrait à une erreur de calcul des poids des bits restants ; 11 ignorerait totalement le rôle du bit de poids fort comme indicateur de signe, en traitant 1011 comme un entier naturel non signé.