

Nom :

Prénom :

Note : / 10

❑ Exercice 1 : Conversions

Compléter le tableau de conversion suivant :

Décimal	Binaire	Hexadécimal
$\overline{205}^{10}$	$\overline{11001101}^2$	\overline{CD}^{16}
$\overline{327}^{10}$	$\overline{101000111}^2$	$\overline{147}^{16}$
$\overline{1068}^{10}$	$\overline{10000101100}^2$	$\overline{42C}^{16}$
$\overline{912}^{10}$	$\overline{1110010000}^2$	$\overline{390}^{16}$
$\overline{2654}^{10}$	$\overline{101001011110}^2$	$\overline{A5E}^{16}$

❑ Exercice 2 : Complément à deux

Dans cet exercice, on suppose que les nombres entiers sont représentés en complément à deux sur 10 bits.

1. Quelle est l'ensemble des nombres représentables ?

L'ensemble des nombres représentables est $\llbracket -2^9; 2^9 - 1 \rrbracket$, c'est à dire $\llbracket -512; 511 \rrbracket$

2. Donner la représentation de
- $\overline{-421}^{10}$

On calcule celle de 421 *sur 10 bits*, on inverse tous les bits, on ajoute 1, on obtient : $\overline{1001011011}^2$

3. Donner la représentation de
- $\overline{-59}^{10}$

De la même façon, on obtient : $\overline{1111000101}^2$ **❑ Exercice 3 : Un programme en C**Le programme C ci dessous compile correctement (et ne produit aucun *warning* avec l'option -Wall) quel sera le résultat de son exécution ? Commenter et justifier.

```

1  #include <stdio.h>
2  #include <stdint.h>
3
4  int main(){
5  for (uint8_t i=10;i>=0;i--)
6  { // %u est le spécificateur de format des entiers non signés
7    printf("%u\n",i);
8  }
9  printf("Partez !\n");
10 }
```

La variable `i` déclarée dans la boucle est de type `uint8_t`, par conséquent elle prend ses valeurs dans $\llbracket 0; 255 \rrbracket$ et un dépassement de capacité n'est pas un comportement indéfini, on effectue simplement les calculs modulo 256. Lorsque la variable `i` atteint zéro, l'opération `i = i - 1` donne `i=255` (puisque $-1 \bmod 256 = 255$). La condition de sortie de boucle est `i<0` et elle n'est donc jamais réalisée, par conséquent, ce programme affiche les entiers de 10 à 0 puis boucle indéfiniment en affichant les entiers de 255 à 0.