

Activité 1 : Dépassement d'entier | 

Dans cette activité et la suivante, les entiers non-signés (entiers positifs) sont codés sur 3 bits. On peut donc représenter les entiers compris entre ___ et _____.

1. Pour rappel, compléter le tableau ci-dessous.

Code			Entier non-signé
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3

2. On donne ci-dessous quatre additions en binaire et leur traduction en décimal dont trois seront complétées par la suite..

$ \begin{array}{r} \\ \\ + \\ \hline 1 \end{array} $	$ \begin{array}{r} \\ \\ + \\ \hline 1 \end{array} $	$ \begin{array}{r} \\ \\ + \\ \hline \end{array} $
$ \begin{array}{r} \\ \\ + \\ \hline \end{array} $	$ \begin{array}{r} \\ \\ + \\ \hline \end{array} $	

(a) Le résultat décimal de la première addition est surprenant : il y a « dépassement d'entier ». Expliquer ce que cela signifie. _____

(b) Compléter les trois autres additions en tenant compte du dépassement d'entier.

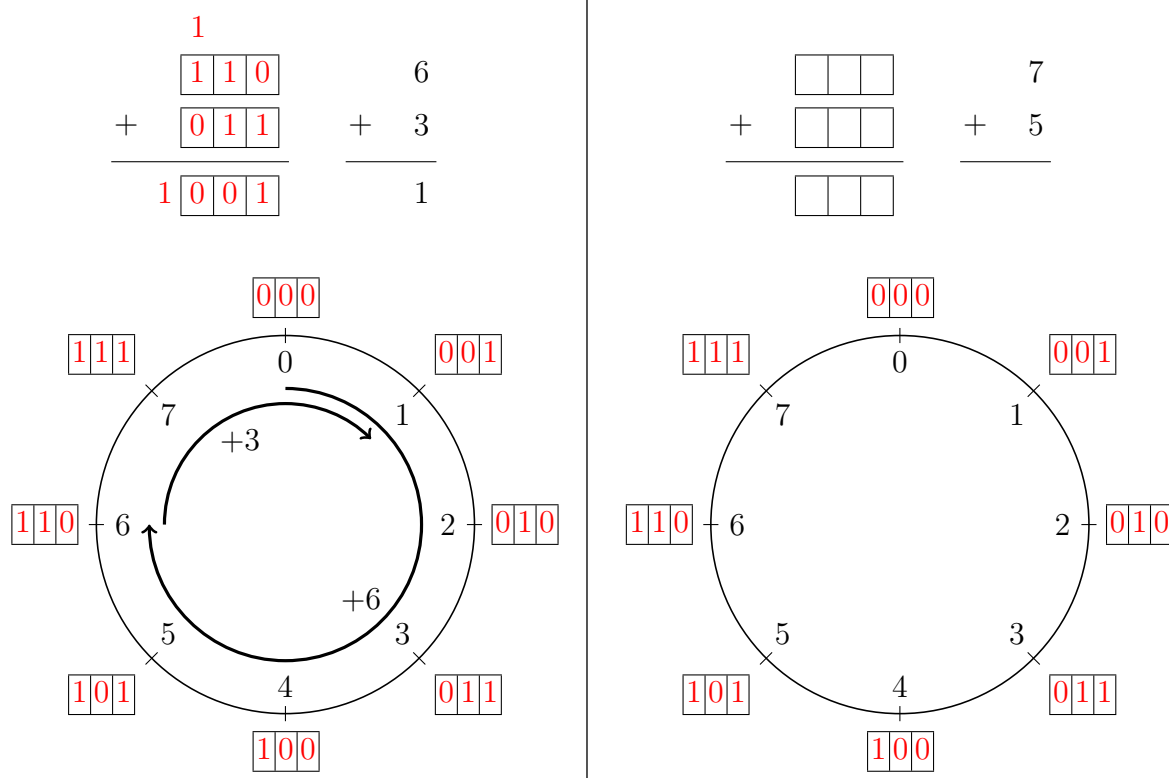
Un peu d'histoire. Le dépassement d'entier le plus célèbre de ces dernières années est très probablement celui qui causa la destruction de la fusée Ariane 5, lors de son vol inaugural, le 4 juin 1996. L'incident, dû à un dépassement d'entier dans les registres mémoire des calculateurs électroniques utilisés par le pilote automatique, a provoqué la panne du système de navigation de la fusée, causant de fait sa destruction ainsi que celle de la charge utile (quatre satellites de la mission *Cluster*, d'une valeur totale de 370 millions de dollars!). L'enquête qui a suivi a montré que le bogue était dû à un module de code obsolète hérité du programme de guidage d'Ariane 4.

Activité 2 : Rudiments d'arithmétique modulaire |

Afin de visualiser les résultats des additions de l'activité précédente, sujettes au dépassement d'entier, on peut s'aider d'un cercle. Voici un exemple ci-dessous à gauche pour l'addition $6 + 3$.

- On part de zéro.
- On avance de 6.
- On continue d'avancer de 3.
- Le point d'arrivée est 1, résultat de l'addition $6 + 3$.

Faire de même pour l'addition ci-dessous à droite.



Remarque. Quel que soit le couple de nombres choisi dans l'ensemble $\{0, 1, 2, 3, 4, 5, 6, 7\}$, le résultat de leur somme sera également un nombre de cet ensemble. On parle d'*arithmétique modulaire*. Le calcul d'une somme s'obtient en prenant le reste par la division par 8 ($2^3 = 8$, la représentation utilisant 3 bits ici.). Exemple :

$$6 + 3 = 9$$

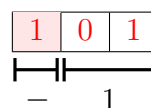
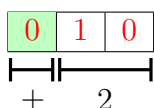
$$9 = 1 \times 8 + 1$$

Le **reste** de la division de 9 par 8 est 1. On note $6 + 3 \equiv 1 \pmod{8}$. On peut retrouver les résultats précédents avec Python à l'aide l'opérateur modulo. Exemples à compléter :

```
>>> (6 + 3) % 8
1
>>> (7 + 5) % 8
...
>>> (5 + 3) % 8
...
>>> (3 + 2) % 8
...
```

Activité 3 : Représentation des entiers signés par signe-valeur absolue | 

Jusqu'à présent nous avons représenté en binaire uniquement des nombres entiers positifs. On aimerait maintenant représenter également les nombres négatifs. Une première solution simple est la représentation par *Signe-Valeur Absolue* (SVA). Cela consiste à sacrifier le bit de poids fort c'est-à-dire le bit le plus à gauche du nombre. Ce premier bit indique le signe du nombre : 0 représente le signe +, 1 représente le signe -. Le reste des bits indique la valeur absolue du nombre. On donne ci-dessous deux exemples d'entiers signés représentés sur 3 bits.



1. Compléter le tableau ci-dessous en respectant les règles de la représentation SVA.

Code			Entier non-signé	Entier signé SVA
0	0	0	0	
0	0	1	1	
0	1	0	2	+2
0	1	1	3	
1	0	0	4	
1	0	1	5	-1
1	1	0	6	
1	1	1	7	

2. Quel premier problème de la représentation SVA peut-on identifier ? _____

3. On souhaite maintenant vérifier si la représentation SVA est compatible avec l'addition. Pour cela, on peut vérifier une règle simple : la somme d'un nombre et de son opposé doit être égale à zéro. Testez ci-après à gauche les deux additions.

4. Les résultats sont décevants. Pour comprendre le problème, compléter le cercle ci-après à droite. Quel second problème de la représentation SVA peut-on identifier ? _____

