

## BASES DE NUMÉRATION

Le projet implémenté sera sauvegardé sous le nom *TP\_08.py*

Lorsqu'on écrit un nombre  $n$  en base  $a$  (avec  $a \geq 2$ ) les seuls chiffres autorisés sont  $\{0; 1; 2; \dots; a-1\}$ . Si  $a \geq 11$ , on complète les chiffres usuels (de 0 à 9) par les lettres A, B, C, etc. Les méthodes vues pour les conversions en bases 10, 2 et 16 se généralisent à un base  $a$  quelconque.

Par exemple, pour convertir 127 en base 3, il suffit de faire les divisions successives de 127 par 3 jusqu'à obtenir un quotient nul. La suite de restes donnera :  $127 = [11201]_3$  (vérifiez!).

Réciproquement, pour convertir  $[5406]_7$  en base 10 il suffit d'écrire (vérifiez!) :

$$[5406]_7 = 6 \times 7^0 + 0 \times 7^1 + 4 \times 7^2 + 5 \times 7^3 = 1917$$

- a) Concevoir la fonction booléenne  $Test(n\_ch, a)$ , à laquelle on transmet un chaîne de caractères  $n\_ch$  et  $a$  un entier, chargé de tester si  $a$  peut représenter un base de numération et si la chaîne transmise peut représenter un entier positif écrit en base  $a$ . Elle renverra Vrai si c'est le cas, et Faux sinon. Par exemple,  $Test("3A57", 6)$  doit renvoyer la valeur Faux et  $Test("3A57", 12)$  doit renvoyer Vrai.
- b) Concevoir la fonction entière  $Conversion\_Decimale(n\_ch, a)$ , à laquelle on transmet un chaîne de caractères  $n\_ch$  représentant un entier écrit en base  $a$ , qui renvoie l'écrit en base 10 de cet entier. On testera la cohérence des valeurs transmises.
- c) Concevoir la fonction chaîne  $Decimale\_Conversion(n, a)$ , à laquelle on transmet un entier  $n$  et un base  $a$ , qui renvoie sous forme de chaîne de caractères l'écrit en base  $a$  de cet entier.
- d) Au final, l'algorithme principal doit :
  - expliquer le rôle de l'algorithme,
  - saisir un base  $a$ ,
  - un nombre  $N\_a\_ch$  écrit dans cette base sous forme de chaîne de caractères,
  - un base  $b$ ,
  - tester la cohérence des valeurs transmises,
  - convertir le nombre donné en base  $b$ ,
  - afficher le résultat en un phrase cohérente.