

# BASE : convertisseur 35 bases

**B**ase est un programme de conversion de bases pour TI z80. Il permet de convertir des nombres de la base 2 à la base 36.

## I. Caractéristiques

Ce programme a été écrit sur TI-83+ en basic. Il fonctionne sans problème sur TI-84+. Il reste à tester sur les autres calculatrices TI z80.

**Mémoire requise** : 414 octets

**Variables utilisées** : A, B, C, D, L, M, N, O, P, R, θ, Str0, Str1

## II. La conversion de bases

Il est facile de changer de système de numération en passant par la base 10 (la base décimale, que nous utilisons habituellement).

Pour convertir un nombre d'une base quelconque vers la base 10, il suffit d'appliquer sur un nombre  $(a_n a_{n-1} a_{n-2} \dots a_0)^b$  écrit en base b la formule suivante :

$$a_n \times b^n + a_{n-1} \times b^{(n-1)} + a_{n-2} \times b^{(n-2)} + \dots + a_0 \times b^0$$

Pour convertir un nombre en base 10 dans une autre base, nous avons un algorithme :

```
N = Nombre en base 10;
C = Base d'arrivée;
D = 0;
I = 1;
tant que N est différent de 0 faire
  D = D + I*mod(N, C);
  N = N / C;
  I = I*10;
fin
/* D est alors le résultat */
```

## III. A propos du programme

L'algorithme précédent ne peut pas être mis en œuvre sur la calculatrice TI car les bases supérieures à 10 sont écrites à la fois avec des chiffres et des lettres. On utilise donc des chaînes pour écrire les nombres. (par exemple, F correspond à 15 en hexadécimal, nous n'avons que 10 chiffres en décimal, les lettres compensent les chiffres manquants).

Il est bien sûr possible de rajouter des caractères (comme les minuscules) pour augmenter le nombre de bases disponibles en conversion. Je ne l'ai pas fait car je juge cela inutile. De plus, il faut savoir que la calculatrice possède 14 chiffres significatifs, ce qui ne lui laisse pas une précision suffisante pour convertir les très grands nombres en valeur exacte.

Si je demande de convertir 1000000000009 en base 10 dans une autre base à mon programme, il commencera par convertir ce nombre en  $10^{13}$ . Le résultat final ne sera donc pas exact mais la conversion de  $10^{13}$ . Comme je pense que la gestion de gros nombres est inutile à cause de l'arrondi, je n'ai laissé que 2 lignes pour écrire un gros résultat.

Enfin, mon programme reconnaît qu'un nombre est écrit dans une base trop grande par rapport à celle indiquée. Ainsi, si vous écrivez 12569 à convertir de la base 8 à 10, sachant qu'il est impossible d'écrire « 12569 » en base 8, le programme vous répondra « ERREUR ».

Pour terminer, une petite astuce : quelle que soit la base dans laquelle vous convertissez, vous trouverez toujours après exécution du programme votre nombre écrit en base 10 dans la variable D :)

Je rappelle que vous êtes libre de modifier ce programme comme bon vous semble, mais mon nom devra être cité dans toute publication l'utilisant.