

## Calculs sur des grands nombres

Licence 1 - Maths Info

---

Afin de traiter des nombres entiers, nous avons vu qu'il existe le type C++ `int`. Il ne permet cependant pas de représenter tous les entiers. Il est généralement possible d'aller jusqu'à 2147483647. Les calculs sur des entiers plus grands vont poser problème, en raison d'un dépassement de la capacité mémoire.

Nous souhaitons dans ce travail construire notre propre type capable de manipuler des entiers très grands. Pour ce faire, nous allons représenter un nombre sous la forme d'un tableau de chiffres. On définit ainsi le type suivant :

```
const int NB_CHIFFRES_MAX=200;
using tabChiffres=array<int, NB_CHIFFRES_MAX>;

struct grandNombre
{
    int nbChiffres;
    tabChiffres c;
};
```

1. Écrire les procédures `afficherNombre` et `lireNombre` permettant de gérer l'affichage et la saisie d'un grand nombre. Nous utiliserons le type `string` pour lire le nombre sous forme d'une chaîne de caractères.
2. Nous souhaitons développer une procédure capable d'additionner 2 grands nombres.
  - (a) Écrire une procédure `additionnerGrandPetit` qui additionne deux grands nombres, en supposant que la taille du premier est supérieure à celle du deuxième.
  - (b) Écrire la procédure `additionner` qui fait la somme de 2 grands nombres sans contrainte sur la taille des nombres.
3. Nous nous intéressons maintenant à la multiplication de grands nombres. Pour ce faire, nous construirons les différentes étapes permettant d'aboutir au résultat.
  - (a) Écrire la procédure `void multiplierParChiffre(const grandNombre &nb, int chiffre, grandNombre &nbm)` qui multiplie *nb* par *chiffre* et met le résultat dans *nbm*.
  - (b) Écrire la procédure `void multiplierParPuissanceDix(grandNombre &nb, int p)` qui multiplie *nb* par  $10^p$ .
  - (c) Écrire la procédure `void multiplier(const grandNombre &nb1, const grandNombre &nb2, grandNombre &nbm)` qui effectue la multiplication de 2 grands nombres.