

TP numéro deux de Maths pour l'info deux

janvier 2015

Exercice 1

Programmer une fonction `multi_pgcd` du type suivant :

```
void multi_pgcd ( const int n , int * a , int d[1] ) ;
```

qui donne un pointeur vers le **pgcd** des nombres `a[0]...a[n-1]`.

Faire des essais.

Exercice 2

Programmer une fonction `puissance` du type suivant :

```
void puissance ( const int n , const int x , const int p , int z[2] ) ;
```

qui donne un vecteur dont la première coordonnée vaut `x` puissance `p` modulo `n` et la deuxième un code de diagnostic qui vaut 0 si tout se passe bien (notamment quand $n > 0$ et $p \geq 0$), les autres valeurs de diagnostic étant à déterminer.

Faire des essais.

Exercice 3

Dans la suite du texte, les nombres entiers naturels sont codés en base 10^9 dans des vecteurs de type `int`. La coordonnée 0 est le chiffre de poids le plus faible. La coordonnée i a pour poids 10^{9i} .

Programmer une fonction `affichage` du type suivant :

```
void affichage ( const int n , int * x ) ;
```

qui affiche en notation décimale l'entier naturel stocké dans le vecteur `x` de dimension `n`.

Faire des essais.

Exercice 4

Programmer une fonction `somme` du type suivant :

```
int somme ( const int n , int * x , const int p , int * y , int * z ) ;
```

qui place dans le vecteur `z` la somme des nombres représentés par les vecteurs `x` et `y`. Les vecteurs `x` et `y` ont respectivement pour dimension `n` et `p`. L'entier retourné par la fonction `somme` est la dimension de `z`.

Faire des essais.

Exercice 5

Construire comme pour l'exercice précédent une fonction `produit` avec le même type que la fonction `somme`.

Faire des essais.