TP2 – Théorème chinois & algorithme de Miller-Rabin

Installation de SageMath.
Ce TP et les suivants utilisent le logiciel SageMath (`http://sagemath.org`), avec le *notebook Jupyter*. Ces logiciels sont installés sur les machines de TP. Si vous voulez l'installer sur une machine personnelle, les consignes sont fournies à l'adresse `https://doc.sagemath.org/html/en/installation/index.html`. En résumé, pour les cas standards :

— Linux : `sudo apt install sagemath sagemath-jupyter` (pour des distributions exotiques, voir le document ci-dessus) ;
— Windows : activer WSL, installer Ubuntu (ou équivalent) et suivre les consignes pour Linux ; éventuellement, possible *via* Cygwin (se référer au document ci-dessus)
— MacOS : installer les binaires du projet 3-manifolds à l'adresse `https://github.com/3-manifolds/Sage_macOS/releases`.

Exercice 1. *Découverte de SageMath*

1. Ouvrir SageMath avec un *notebook* Jupyter, qui doit s'ouvrir dans un onglet du navigateur. *Sous linux, depuis un terminal, exécuter* `sage -n jupyter`. *Sous MacOS, lancer l'application et sélectionner le notebook.*
2. Créer un nouveau document (bouton « Nouveau ») de type SageMath 9.x.
3. En cliquant sur son nom par défaut (« Untitled » *a priori*), le renommer avec le nom de votre choix. Repérer l'extension donnée automatiquement au fichier.
4. SageMath peut (presque) être vu comme une bibliothèque de Python : vérifier dans des *cellules* de calcul que vous pouvez taper le code Python que vous voulez. *On exécute une cellule avec* <Shift>+<Entrée>.
5. SageMath n'est pas qu'une bibliothèque, il introduit quelques différences de type ou de syntaxe. Par exemple, les entiers que vous entrez sont de type ZZ et non `int`. Définir une variable `n` avec une valeur entière, puis écrire « `n.` » et appuyer sur <tab> pour voir toutes les méthodes rattachées à un entier. Chercher celle qui vous semble calculer un PGCD, et vérifier avec quelques exemples.
6. Vérifier le type des entiers renvoyés par la fonction `range` (qui vient de Python). Comparer avec la fonction `srange` (fournie par SageMath).
7. Pour accéder à l'aide d'une méthode (par exemple la méthode `xgcd` d'un entier `n`), on écrit `n.xgcd?` et on exécute la cellule. Que fait `n.xgcd(...)` ?

Exercice 2. *Théorème chinois*

1. Le théorème chinois, et l'algorithme RESTESCHINOIS, utilisent des PGCD et des inversions modulaires. Trouver la méthode sur les entiers qui étant donné a et N calcule l'inverse de a modulo N.
2. Écrire une fonction `ResteChinois` qui prend en entrée deux listes ($n_1, \dots, n_k$ et $a_1, \dots, a_k$) et renvoie l'unique entier $z < \prod_i n_i$ qui vérifie $z \equiv_{n_i} a_i$ pour tout i. *Votre fonction doit vérifier que les entrées sont correctes, c'est-à-dire que les deux listes ont la même longueur, et que les n_i sont premiers deux à deux.*
3. SageMath fournit des fonctions qui permettent d'écrire facilement du code mathématique. En particulier, lire la documentation de la fonction `prod` et écrire une nouvelle version de `ResteChinois` qui l'utilise. On peut aussi utiliser la fonction `sum` de la même façon.
4. Le théorème chinois permet de définir une bijection entre $\{0,\dots,n_1-1\}\times\cdots\times\{0,\dots,n_k-1\}$ et $\{0,\dots,N-1\}$ où $N = \prod_{i=1}^{k} n_i$. Quel sens de la bijection est fourni par `RestesChinois` ? Écrire la fonction réciproque, et tester que les deux fonctions sont bien réciproques l'une de l'autre.

Exercice 3. *Algorithme de Miller-Rabin*
Télécharger le fichier `MillerRabin.ipynb` et suivre les consignes.