	ANNÉE UNIVERSITAIRE 2022/2023 **4TMA701U Calcul Formel** **Devoir Surveillé** **Date : 09/11/2022 Heure : 15h30 Durée : 1h30** Documents non autorisés.	**Collège Sciences et Technologies**

Vous rendrez à la fin de l'examen une copie papier ainsi qu'un fichier sage contenant vos programmes (lisible, commenté et nettoyé si possible..) au format DS-Nom-Prenom.ipynb (feuille Jupyter) ou DS-Nom-Prenom.sage (fichier texte). Le fichier est à envoyer par e-mail à votre enseignant.e de TD (christine.bachoc@u-bordeaux.fr ou gilles.zemor@u-bordeaux.fr).

Exercice 1 Soient $p_1 < p_2 < \cdots < p_m < p_{m+1} < p_{m+2}$ une suite strictement croissante de $m+2$ nombres premiers. On note M le produit des m premiers termes de la suite, soit $M = p_1p_2\cdots p_m$.

Dans toute la suite, pour un entier x, et pour tout $i \in [1, \dots, m+2]$, on note x_i son reste modulo p_i.

1. Soit x un entier tel que $0 \leq x < M$. Montrez que si on vous donne m parmi les $m+2$ valeurs des x_i, alors vous pouvez reconstituer x sans ambiguïté (expliquez comment et justifiez votre réponse).

   Indication : Commencez par traiter le cas où les m valeurs données sont les m premières et pensez à utiliser un célèbre théorème d'arithmétique ..

   Exemple numérique : $(p_1, p_2, p_3, p_4, p_5, p_6, p_7) = (2, 3, 5, 7, 11, 13, 17)$, avec $(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (*, 0, 2, 6, 9, *, 16)$, les "*" représentant les valeurs manquantes. Que vaut x ?

   Vous pouvez utilisez la fonction **crt** *de sage ; expliquez votre algorithme et écrivez une fonction qui prend en entrée une liste de premiers et une liste de restes dans laquelle les deux restes manquent (remplacez-les par exemple par des -1), et sort x.*

2. Soit x un entier tel que $0 \leq x < M$. On vous donne maintenant les $m+2$ restes modulo p_i de x, mais une de ces valeurs, vous ne savez pas laquelle, est fausse. Dit autrement, on dispose de $(y_1, y_2, \dots, y_{m+1}, y_{m+2})$ avec $y_i = x_i$ pour toutes les valeurs de i *sauf une*. Montrez que vous pouvez retrouver l'entier x sans ambiguïté et expliquez comment.

   Exemple numérique : les mêmes p_i que précédemment, avec $(y_1, y_2, y_3, y_4, y_5, y_6, y_7) = (1, 1, 3, 3, 5, 3, 8)$. Que vaut x ?

   Vous pouvez utilisez la fonction **crt** *de sage ; expliquez votre algorithme et écrivez une fonction qui prend en entrée une liste de premiers et une liste de restes, et sort x.*

Exercice 2 Dans cet exercice vous allez étudier une variante du test de primalité de Pocklington Lehmer vu en cours.

Dans tout l'exercice, n est un entier impair, tel que $n-1=pu$ avec p un nombre premier impair, $p>\sqrt{n}$, et $\text{pgcd}(p,u)=1$. On considère l'hypothèse (H) suivante :

$$\text{(H)}\quad \text{Il existe un entier } b,\ 1\leq b<n \text{ tel que : } \begin{cases} b^{n-1}=1 \bmod n \\ \text{pgcd}(b^u-1,n)=1 \end{cases}$$

1. Dans cette question vous allez montrer que si (H) est vérifiée alors n est premier. On suppose donc (H).
   a) Supposons que n a un diviseur premier q avec $q\leq\sqrt{n}$. Soit $c=b^u \bmod q$. Montrez que $c\neq 1 \bmod q$ mais que $c^p=1 \bmod q$.
   b) En déduire que n est premier.
2. Déduire de la question 1) un test de primalité pour n, prenant en entrées n, p et b comme ci-dessus et sortant "n est premier" ou "n est composé" ou "échec, on ne peut pas conclure". Programmez ce test sous la forme d'une fonction sage.
   Indication : Attention de ne pas faire intervenir dans l'exécution des entiers plus grands que n. Vous pouvez utiliser la fonction **IntegerModRing()** .
3. Analysez l'ordre de grandeur de la complexité binaire de votre test en fonction de la taille binaire de n.
4. Construire une liste de nombres premiers de plus en plus grands en partant de $p_1 = 1000003$ et en cherchant grâce à votre test de primalité avec $b=2$, pour $i\geq 1$, un nombre premier p_{i+1} de la forme $p_i(10^{e_i-1}+k)+1$, où $10^{e_i}\leq p_i<10^{e_i+1}$ (essayer successivement $k=2,4,6,\ldots$). Vous devriez pouvoir dépasser 100 chiffres décimaux.