Toute réponse non justifiée sera considérée comme vide. À l'inverse, tout raisonnement, même non abouti, sera valorisé. Toute question non traitée pourra être admise pour usage ultérieur. **Ce sujet comporte 3 pages.**

Les programmes demandés sont à écrire en langage `python`*, avec une syntaxe précise et une indentation claire. Toute rédaction approximative ou ambiguë sera sanctionnée. On pourra faire usage de l'aide mémoire suivante et supposer le module* `numpy` *importé*

▶ **Opérations**

- `a%b` renvoie le reste de la division euclidienne de `a` par `b` ;
- `a**b` renvoie $\mathtt{a}^{\mathtt{b}}$.

▶ **Manipulation de la classe `list`**

- `len(u)` renvoie la longueur de la liste `u` ;
- `u+v` concatène les deux listes `u` et `v` ;
- `u.append(e)` ajoute l'élément `e` à la fin de la liste `u` ;
- `e in u` teste si l'élément `e` est dans la liste `u`.

♦ Exercice 1 : fonctions indicatrices

Dans tout ce problème, on fixe un ensemble E quelconque. On rappelle que si $A \in \mathcal{P}(E)$, l'indicatrice de A est la fonction suivante :

$$\begin{aligned} \mathbb{1}_A : E &\to \{0,1\} \\ x &\mapsto \begin{cases} 1 \text{ si } x \in A \\ 0 \text{ si } x \notin A \end{cases} . \end{aligned}$$

1. On considère l'ensemble $\mathcal{F} = \mathcal{F}(E, \{0,1\})$ des applications de E dans $\{0,1\}$.
   (a) Lister les éléments de $\mathcal{F}$ dans le cas où $E = \{0,1\}$.
   (b) Démontrer que si E contient strictement plus de deux éléments, $\mathcal{F}$ ne contient aucune bijection.
2. (a) Soient $A, B \in \mathcal{P}(E)$; démontrer que $\mathbb{1}_A = \mathbb{1}_B \Leftrightarrow A = B$.
   (b) Soit $f \in \mathcal{F}$; montrer qu'il existe $A \in \mathcal{P}(E)$ tel que $f = \mathbb{1}_A$.
   (c) En déduire que l'application

$$\begin{aligned} \varphi : \mathcal{P}(E) &\to \mathcal{F} \\ A &\mapsto \mathbb{1}_A \end{aligned}$$

   est une bijection.

♦ Exercice 2 : calculs de sommes

L'objectif de cet exercice est de calculer l'entier

$$\begin{aligned} N &= \sum_{k=1}^{999} k(k+1)(k+2) \\ &= (1 \times 2 \times 3) + (2 \times 3 \times 4) + (3 \times 4 \times 5) + \ldots + (999 \times 1000 \times 1001) \end{aligned}$$

de deux façons différentes.

1. On fixe $n \geq 2$.
   (a) Déterminer (en le démontrant) la valeur de la somme $\sum_{k=0}^{n} k$.

(b) Démontrer que :

$$\sum_{k=0}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2 .$$

(c) En déduire la valeur de N. ***Indication :*** *on pourra s'intéresser à la somme* $\sum_{k=0}^{n} k^3 - k$.

2. On pose, pour tout $n \geq 2$:

$$\begin{aligned} N_n &= \sum_{k=1}^{n-1} k(k+1)(k+2) \\ &= (1\times 2\times 3) + (2\times 3\times 4) + (3\times 4\times 5) + \ldots + (n-1)n(n+1) . \end{aligned}$$

(a) Démontrer que, pour tout $n \geq 2$, on a :

$$N_n = \frac{(n-1)n(n+1)(n+2)}{4}$$

(b) En déduire la valeur de N.

⧫ Problème : extrait CAPES 2018

On souhaite dans ce problème crypter des messages, lettre par lettre. Pour les écrire, on utilise 29 caractères différents : les 26 lettres de l'alphabet et les trois symboles virgule (,), espace (␣) et point (.). Pour faciliter la mise en place du problème, on associe à chacun de ceux-ci un entier, selon le tableau suivant.

.	␣	A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

N	O	P	Q	R	S	T	U	V	W	X	Y	Z	,
15	16	17	18	19	20	21	22	23	24	25	26	27	28

On note $\mathcal{R}$ l'ensemble $[\![0, 28]\!]$ des entiers utilisés ici. Pour $k \in \mathbb{N}^*$, on note f_k l'application de $\mathcal{R}$ dans lui–même associant à chaque $x \in \mathcal{R}$ le reste de la division euclidienne de x^k par 29. On appellera ***codage*** *associé à f_k le procédé consistant à remplacer tous les éléments de $\mathcal{R}$ par leur image par la fonction f_k. Par exemple, la lettre E correspondant à l'entier 6 ; elle sera remplacée par le symbole correspondant à l'entier $f_k(6)$.*

– I –

1. Coder le message "AU SECOURS" à l'aide de la fonction f_3.
2. (a) Justifier que pour tous $k \in \mathbb{N}^*$ et $x \in \mathcal{R}$, $f_k(x) \equiv x^k[29]$.
   (b) Démontrer que, quelque soit le codage choisi, les symboles espace et point restent inchangés.
3. Écrire une fonction `python` d'entête `codage(k,L)` prenant en entrée un entier non nul `k` et une liste `L` d'entiers de $\mathcal{R}$ représentant un message et renvoyant la liste d'entiers correspondant au codage du message par la fonction f_k.
4. (a) En appliquant cet algorithme, on obtient le tableau de codage suivant pour la fonction f_7.

.	␣	A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	12	12	28	28	28	1	17	28	17	12	17	28	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z	,
17	1	12	17	12	1	12	28	1	1	1	17	17	28

   Expliquer, succinctement mais efficacement, pourquoi ce codage présente un **gros** problème.

   (b) Quelle propriété la fonction f_k doit–elle vérifier pour que le codage associé soit utilisable en pratique ?

– II –

1. On admet que, pour tout $0 < k < 29$, 29 divise le coefficient binomial $\binom{29}{k}$.
   (a) Montrer que pour tous $a, b \in \mathbb{Z}$, $(a+b)^{29} \equiv a^{29} + b^{29}$ [29].
   (b) Démontrer par récurrence sur $a \in \mathbb{N}$ que
   $$\forall\, a \in \mathbb{N},\ a^{29} \equiv a[29]\ .$$
   (c) Généraliser (avec démonstration) ce résultat au cas où $a \in \mathbb{Z}$.
2. Décrire la fonction f_{29}.

– III –

Les programmes demandés sont à écrire en langage `python`*, avec une syntaxe précise et une indentation claire. Toute rédaction approximative ou ambiguë sera sanctionnée.*

1. (a) Écrire une fonction d'entête `Transfo(c)` prenant en entrée une chaîne de caractères `c` et renvoyant la liste d'entiers correspondant (en respectant le tableau initial).
   (b) Écrire une fonction d'entête `Codage(k,c)` prenant en entrée un entier non nul `k` et une chaîne de caractère c et renvoyant la liste d'entiers correspondant au codage du message par la fonction f_k. On pourra évidemment faire usage de fonctions précédemment écrites.
2. Écrire une fonction d'entête `SansDoublons(L)` prenant en entrée une liste L d'entiers et renvoyant `True` si elle est composée d'entiers tous distincts et `False` sinon.
3. Écrire une fonction d'entête C`odageValide(k)` prenant en entrée un entier `k` non nul et renvoyant `True` si l'application f_k permet de coder sans ambiguïté un message (*cf.* **I.4.b**), `False` sinon.