DS 4 – Maths & Info

La calculatrice est interdite.
On pourra admettre une question (en le précisant) pour poursuivre la résolution des exercices.
La qualité de la rédaction, la clarté et la précision des raisonnements seront pris en compte dans l'évaluation, ainsi que le soin.

Exercice 1. Systèmes

1. Résoudre
$$\begin{cases} x + y + 2z = 5 \\ x - y - z = 1 \\ x + z = 3 \end{cases}$$
2. Résoudre en fonction de λ le système :
$$\begin{cases} -\lambda x + 3y + 2z = 0 \\ x - \lambda y = 0 \\ y - \lambda z = 0 \end{cases}$$

Exercice 2. Un produit

Pour tout entier naturel n, on pose
$$u_n = \prod_{k=1}^{n} \left(1 + \frac{1}{2^k}\right)$$

1. Calculer u_0, u_1, u_2 et u_3. On donnera les résultats sous la forme d'entiers ou de fractions irréductibles.
2. Exprimer u_{n+1} en fonction de u_n.
3. Montrer que pour tout $n \in \mathbb{N}^*$, on a $u_n \geq \frac{3}{2}$.
4. a) Montrer que pour tout $x \in]-1; +\infty[$: $\ln(1+x) \leq x$
   b) En déduire un majorant de $\ln(u_n)$, puis de u_n.

Exercice 3.

Soit $n \in \mathbb{N}$.

1. **Python**
   a) Définir une fonction `fact(n)` prenant comme argument un entier et renvoyant $n!$
   b) Définir une fonction `binom(n,k)` prenant en argument deux entiers et renvoyant $\binom{n}{p}$
2. Soit $k \leq n$, montrer que :
$$\frac{1}{k+1}\binom{n}{k} = \frac{1}{n+1}\binom{n+1}{k+1}$$
3. A l'aide de la formule du binôme, montrer que :
$$\sum_{k=0}^{n} \binom{n+1}{k+1} = 2^{n+1} - 1$$
Indication : On pourra effectuer le changement d'indice : $i = k + 1$
4. En déduire la valeur de :
$$S_1 = \sum_{k=0}^{n} \frac{1}{k+1}\binom{n}{k}$$

5. On pose :
$$S_2 = \sum_{k=0}^{n} \frac{(-1)^k}{k+1}\binom{n}{k}$$
Montrer que $S_2 = \dfrac{1}{n+1}$

Exercice 4.

Soit $n \in \mathbb{N}$, on pose :
$$u_n = \sum_{j=0}^{n} \sum_{k=j}^{n} 2^k$$

1. **Python** Écrire le script d'une fonction `somme(n)` renvoyant u_n.
2. Soit $(p,q) \in \mathbb{N}^2$, tel que $p \leq q$, montrer que $\displaystyle\sum_{k=p}^{q} 2^k = 2^{q+1} - 2^p$
3. En déduire que $u_n = n2^{n+1} + 1$
4. En permutant les sommes, montrer que
$$u_n = \sum_{k=0}^{n} (k+1)2^k$$
5. Déduire des questions précédentes que :
$$\sum_{k=0}^{n} k2^{k-1} = (n-1)2^n + 1$$
6. A l'aide des question précédentes, calculer :
$$S_n = \sum_{i=0}^{n} \sum_{k=0}^{i} (k+1)2^k$$
7. En déduire la valeur de
$$T_n = \sum_{k=0}^{n} k(k+1)2^k$$
On exprimera T_n en fonction S_n.
8. Calculer
$$\sum_{k=0}^{n} k^2 2^k$$
Indication : On pourra écrire $k^2 = k(k+1) - k$

Exercice 5. Python

1. Écrire une fonction `maximum(L)` qui prend en argument une liste de nombres et renvoie le maximum de tous les éléments de la liste.
2. Pour une liste L, la liste miroir est la liste contenant les mêmes éléments mais rangés dans l'ordre inverse. Écrire la fonction `miroir(L)` qui renvoie la liste miroir de L.
3. Une liste L est symétrique si elle est égale à sa liste miroir, par exemple $[4, 5, 2, 5, 4]$ ou bien $[3, 7, 7, 3]$.

   Écrire une fonction `est_symétrique(L)` qui renvoie `True` si la liste `L` est symétrique et `False` sinon.
4. Une liste de nombres L est dite pyramidale si et seulement si elle vérifie les trois conditions suivantes :
   - Elle est de longueur impaire,
   - Elle est symétrique,
   - Elle est strictement croissante sur sa première moitié (et donc automatiquement strictement décroissante sur sa deuxième moitié).

   C'est le cas par exemple de $[1, 3, 8, 12, 8, 3, 1]$.
   Écrire une fonction `est_pyramidale(L)` qui renvoie `True` si `L` est une liste pyramidale, et `False` sinon.