Programme de colle 7
Semaine du 9 novembre 2026

Chapitre 4 : Sommes et produits

Note aux colleurs : nous avons vu la notation $\prod$ pour les produits. Merci de poser également une question de calcul de somme avec changement de variable ou relation de Chasles

➢ **Cours à connaître**

- ✔ Connaitre parfaitement les formules pour calculer les sommes usuelles $\sum_{k=0}^{n} a$ (avec a une constante) ; $\sum_{k=0}^{n} k$; $\sum_{k=0}^{n} k^2$; $\sum_{k=0}^{n} q^k$
  On peut aussi connaitre partant d'un rang p afin de gagner du temps dans les calculs (valable pour les sommes de la forme $\sum a$, $\sum k$ et $\sum q^k$)
- ✔ Connaître les propriétés des sommes : relation de Chasles, linéarité et inégalité triangulaire.
- ✔ Connaître la définition de $n!$ et la propriété $n! = n \times (n-1)!$ (ou $(n+1)! = (n+1)n!$).
- ✔ Connaître les formules de calcul de produit : $\prod_{k=1}^{n} k = k!$ et $\prod_{k=p}^{n} a = a^{n-p+1}$

➢ **Exercices type**

- ✔ Calculer une somme avec les formules usuelles, la relation de Chasles ou un changement d'indice
- ✔ Calculer un produit (peu de formules, on peu utiliser la relation de Chasles et la « non linéarité » pour faire « sortir » une constante)
- ✔ Exercices plus longs et guidés :
  - — Transformer une somme en produit et réciproquement avec ln et exp.
  - — Prouver des sommes et produits d'inégalités.

Chapitre 5 : Récurrence

➢ **Cours à connaître**

- ✔ Bien connaitre les étapes d'un raisonnement par récurrence simple
- ✔ Connaitre la récurrence double (penser à la double initialisation)

➢ **Exercice types**

- ✔ Prouver l'expression du terme général d'une suite définie par une relation de récurrence. Par exemple :
  Soit (u_n) la suite définie par $u_0 = 3$ et pour tout $n \in \mathbb{N}$, $u_{n+1} = 5u_n - 4$. Montrer par récurrence que pour tout $n \in \mathbb{N}$, $u_n = 1 + 2 \times 5^n$.
- ✔ Prouver une relation sur une suite définie par une relation de récurrence. Par exemple, prouver que $0 \leq u_n \leq 1$ pour $n \in \mathbb{N}$.
- ✔ Prouver une relation sur des entiers. Par exemple :
  Montrer que la proposition $\left(\forall n \geq 4,\ 2^n \geq n^2\right)$ est vraie.

- ✔ Prouver une formule de somme. Par exemple :
  Montrer que pour tout $n \in \mathbb{N}$, on a $\sum_{k=0}^{n} k = \frac{n(n+1)}{2}$.
- ✔ Utiliser une récurrence double. Par exemple :
  On considère la suite (u_n) définie par $u_0 = 1$, $u_1 = 0$ et pour $n \in \mathbb{N}$, $u_{n+2} = 4u_{n+1} - 4u_n$. Montrer par récurrence double que pour tout $n \in \mathbb{N}$ on a $u_n = 2^n - n2^{n-1}$.

Chapitre 6 : Généralités sur les suites

➢ **Cours à connaître**

- ✔ Définition d'une suite
- ✔ Définition d'une suite bornée, majorée, minorée (et les négations).
- ✔ Définition d'une suite croissante, d'une suite décroissante, d'une suite monotone, d'une suite constante.

➢ **Exercices type**

- ✔ Calculer les premiers termes d'une suite (définie explicitement ou par une relation de récurrence)
- ✔ Savoir représenter graphiquement une suite (un seul exemple vu en cours).
- ✔ Montrer qu'une suite est bien définie
  Ex : Soit (u_n) définie par $u_0 = 5$ et $u_{n+1} = \frac{1}{1+u_n}$ pour $n \in \mathbb{N}$. Montrer que (u_n) est bien définie. On utilise une récurrence pour montrer que $u_n \neq -1$)
- ✔ Prouver par récurrence (simple ou double) le terme général d'une suite définie par récurrence.
- ✔ Montrer qu'une suite est bornée, majorée ou minorée ou non (utiliser soit une preuve directe (suite définie explicitement) soit une récurrence (suite définie par une relation de récurrence)
- ✔ Étudier le sens de variations d'une suite :
  - → en étudiant le signe de $u_{n+1} - u_n$
  - → en comparant $\frac{u_{n+1}}{u_n}$ à 1 (valable si $u_n > 0$ pour tout n)
  - → en étudiant les variations de f si $u_n = f(n)$
  - → par récurrence si $u_{n+1} = f(u_n)$ avec f croissante.

Python : Boucles `for`

- ✔ Connaître la structure d'une boucle for (en particulier bien écrire les `:` en fin de ligne et respecter l'indentation)
- ✔ Connaître les instructions `range` sous les formes `range(b)`, `range(a,b)` ou `range(a,b,pas)`

Question de cours / 2 : On peut demander d'écrire une instruction `range`, par exemple pour produire les entiers pairs entre 20 et 100. Ou proposer un programme à interpréter :

```
for i in range(5) :
        print(3*i)
```

Exercice /3 : Écrire un programme utilisant une boucle `for`. Par exemple :
• Écrire un programme qui demande à l'utilisateur de saisir un entier n puis affiche la table de multiplication correspondante (par exemple en saisissant `5` on doit afficher les nombres `5 ; 10 ; 15 ; ... 50`.
• Écrire un programme qui affiche les 10 premières puissances de 2. On affichera donc `1 ; 2; 4 ;... 1024`.
• Pour ceux qui sont à l'aise, on peut demande un calcul de somme (calculer la somme des entiers de 1 à 100 ou de 1 à n). Non vu encore en classe.