Programme de colle 5
Semaine du 12 octobre 2026

Chapitre 4 : Sommes et produits

Attention : nous n'avons pas du tout abordé les produits avec le symbole $\prod$. Les étudiants doivent connaitre la définition d'une factorielle (en écrivant avec des pointillés) et simplifier en décomposant.

➢ **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 $(n+1)! = (n+1)n!$).

➢ **Exercices type**

- ✔ Passer de l'écriture avec le symbole $\sum$ à une écriture avec les pointillés et inversement
- ✔ Calculer une somme se ramenant à une somme usuelle (par exemple $\sum_{k=0}^{n}(2k+1)^2$ en développant ; $\sum_{k=1}^{n} 2^k \times 3^{2k}$, en écrivant d'abord $2^k \times 3^{2k} = (2 \times 3^2)^k$
- ✔ Pour se ramener à une somme usuelle :
  - → Citer la linéarité quand on l'utilise
  - → Utiliser un changement d'indice pour simplifier l'expression
  - → Utiliser la relation de Chasles si les indices ne correspondent pas à une somme connue (ex : $\sum_{k=4}^{n} k^2$)
- ✔ Calculer une somme avec un télescopage.
- ✔ Simplifier des expressions avec des factorielles. Penser à utiliser le lien entre $n!$ et $(n-1)!$ ou entre $n!$ et $(n+1)!$. Ex : simplifier $\dfrac{200!}{202!}$ ou $\dfrac{(n+2)!}{n!}$

Il n'est pas attendu que les élèves récitent par coeur les formules de changement d'indices ou de télescopage. En revanche, ils doivent maîtriser la méthode.

Chapitre 5 : Récurrence

➢ **Cours à connaître**

✔ Bien connaitre les étapes d'un raisonnement par récurrence simple

➢ **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 $n \in \mathbb{N}$, $u_n = 1 + 2 \times 5^n$.
Seul un exemple de ce type a été travaillé en classe, les étudiants peuvent être guidés pour tout autre exercice.

✔ 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 $\displaystyle\sum_{k=0}^{n} k = \frac{n(n+1)}{2}$.

Note aux colleurs. La récurrence double n'a pas encore été vue en cours.

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.