Programme de colle n^o 5 : Sommes et produits. Généralités sur les suites réelles.

*Semaine du lundi **13 octobre**.*

Le programme de la semaine précédente est toujours au programme de cette semaine.

Sommes et produits (fin)

5.1 Factorielle d'un entier. Propriétés dites "de morphisme" de l'exponentielle et du logarithme pour les sommes et produits finis. Sommation d'inégalités, produits d'inégalités à membres tous positifs. Produit et valeur absolue. Inégalité triangulaire généralisée aux sommes finies.

5.2 Exemples de double sommes, notations liées aux double sommes, permutation des sommes.

Généralités sur les suites réelles

5.3 Notion de suite réelle. Ensemble d'indexation. Ensemble $\mathbb{R}^{[\![n_0,+\infty[\![}$ des suites réelles indexées par $[\![n_0,+\infty[\![$ (où $n_0 \in \mathbb{N}$). Définition d'une suite $(u_n)_{n\geq n_0}$ par une formule donnant u_n pour tout $n \geq n_0$. Définition d'une suite $(u_n)_n$ par récurrence et exemples de justifications de la bonne définition de u_n pour tout n convenable. Suites définies de manière implicite : exemples.

5.4 Représentation graphique d'une suite $(u_n)_n$ en plaçant les points (n, u_n). Représentation graphique propre aux suites définies par une relations de récurrences de la forme "$u_{n+1} = f(u_n)$", à l'aide du tracé du graphe de f et de la droite d'équation $y = x$.

5.5 Suites majorées, minorées, bornées. Caractérisation des suites bornées à l'aide de la valeur absolue.

5.6 Suites monotones et strictement monotones, suites constantes.

Python

5.7 Fonctions. Premiers algorithmes sur les listes (simples utilisation des syntaxes `L[i]`, `len(L)`,`L.append(...)` et des boucles `for` pour remplir une liste, aucun algorithme plus complexe n'a été vu jusqu'ici).

Quelques questions de cours

1. Soient $n \in \mathbb{N}^*$ et $(u_k)_{k\in\mathbb{N}}$ une suite réelle. Compléter et démontrer (démo avec points de suspension) : $\displaystyle\sum_{\substack{k=0 \\ k \text{ impair}}}^{2n} u_k = \ldots$.

   Notons $P = \displaystyle\prod_{\substack{k=1 \\ k \text{ pair}}}^{2n+1} k$ et $I = \displaystyle\prod_{\substack{k=1 \\ k \text{ impair}}}^{2n+1} k$. Calculer P et en déduire I.
2. Énoncer les propriétés calculatoires liées au symbole $\prod$ (prop. 43). Énoncer et démontrer, à l'aide de ces propriétés, les formules donnant $\displaystyle\prod_{k=p}^{n} \lambda a_k$ ($\lambda \in \mathbb{R}$ fixé) et $\displaystyle\prod_{k=p}^{n} \frac{a_k}{b_k}$.
3. Énoncer les propriétés "de morphisme" de l'exponentielle et du logarithmes généralisées aux sommes et produits finis. Démontrer celle du logarithme.
4. Énoncer et démontrer l'inégalité triangulaire (prop. 56 (ii)) pour les sommes finies.
5. Soit $n \in \mathbb{N}^*$. Calculer $\displaystyle\sum_{1\leq i,j\leq n} \min(i,j)$.
6. Définir la notion de suite réelle. Soit u la suite donnée par $u_0 = \dfrac{3}{4}$ et : $\forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n(1-u_n)}$. Montrer que u_n est bien défini pour tout entier naturel n.
7. Notons, pour tout $n \in \mathbb{N}$, (E_n) l'équation $x^n + x - 1 = 0$. Montrer que, pour tout $n \in \mathbb{N}$, (E_n) admet une unique solution sur $\mathbb{R}_+$, que l'on note u_n. Montrer que : $\forall n \in \mathbb{N}, u_n \in [\dfrac{1}{2}, 1]$.
8. Définir la notion de suite minorée. Énoncer et démontrer la caractérisation des suites bornées utilisant la valeur absolue. *Élèves : Version ajustée de l'énoncé sur cahier de prépa.*
9. Écrire le code d'une fonction d'entête `def estCarre(n):` prenant en entrée un entier naturel `n` et renvoyant en sortie `True` si `n` et le carré d'un entier, et `False` sinon.
10. Écrire le code d'une fonction Python d'entête `def GardePositifs(L):` prenant en entrée une liste de nombres `L` et renvoyant en sortie la liste obtenue à partir de `L` en ne gardant que ses éléments positifs.