Semaine n°11 du 11 décembre au 15 décembre

Informatique(Python) : cf exemples en annexe

- Boucle `while` + Compteur.
- Fonction récursive
- Boucle `for` Calcul de somme.

Méthodes de calcul : sommes et produits

- Notation $\sum$: définition, linéarité, Chasles, changement d'indice.
- Sommes téléscopiques.
- Sommes classiques à connaître
  - $\sum\limits_{k=1}^{n} 1 = n, \quad \sum\limits_{k=0}^{n} 1 = n+1, \quad \sum\limits_{k=i}^{j} 1 = j-i+1, \quad \sum\limits_{k=i}^{j} a = (j-i+1)a$ où $a \in \mathbb{C}$
  - $\sum\limits_{k=0}^{n} k = \dfrac{n(n+1)}{2}, \quad \sum\limits_{k=0}^{n} k^2 = \dfrac{n(n+1)(2n+1)}{6}$
  - $\forall q \in C, q \neq 1, \sum\limits_{k=0}^{n} q^k = \dfrac{1-q^{n+1}}{1-q}$ et de manière générale $\sum\limits_{k=i}^{j} q^k = q^i \dfrac{1-q^{j-i+1}}{1-q}$
- Sommes doubles du type $\sum\limits_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} a_{ij}$ et $\sum\limits_{1 \leq i \leq j \leq n} a_{ij}$.
- Produits : définition, notation $\prod$, propriétés, $\prod\limits_{k=i}^{j} c$ où $c \in \mathbb{C}$
- Coefficients binomiaux :
  - Factorielle
  - Définition $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ pour $k \in [\![0, n]\!]$
  - Formules : $\binom{n}{0}, \binom{n}{1}$, symétrie $\binom{n}{k} = \binom{n}{n-k}$, formule "sans nom" $\binom{n+1}{k+1} = \frac{n+1}{k+1}\binom{n}{k}$
  - Triangle de Pascal
  - Binôme de Newton

Nombres complexes

- Forme algébrique d'un nombre complexe, somme, produit, partie réelle et imaginaire, représentation géométrique.
- Conjugué :définition, interprétation géométrique, propriétés, caractérisation des nombres réels et des nombres imaginaires purs avec le conjugué.
- Module ($|z| = \sqrt{x^2+y^2}$), expression avec le conjugué ($|z| = \sqrt{z\bar{z}}$), interprétation géométrique, propriétés
- Inégalités triangulaires :
  $\forall (z, z') \in \mathbb{C}^2, \qquad |z+z'| \leq |z| + |z'|$
  $\forall (z, z') \in \mathbb{C}^2, \qquad ||z| - |z'|| \leq |z - z'|$
- Notation $e^{i\theta}$, propriétés $e^{i\theta} \times e^{i\theta'} = e^{i(\theta+\theta')}$ et $\frac{e^{i\theta}}{e^{i\theta'}} = e^{i(\theta-\theta')}$, formule de Moivre, formules d'Euler.
- Linéarisation d'expressions trigonométriques.
- Argument d'un nombre complexe non nul, mise sous forme exponentielle d'un nombre complexe (technique de l'angle moitié), égalité de deux complexes sous forme exponentielle :

$$\forall (r, r', \theta, \theta') \in (\mathbb{R}_+^*)^2 \times \mathbb{R}^2, \qquad r\, e^{i\theta} = r'\, e^{i\theta'} \iff \begin{cases} r = r' \\ \theta \equiv \theta' [2\pi] \end{cases}$$

- Résolution des équations du second degré à coefficients réels, somme et produit des solutions. Résolution de l'équation $x^2 = a$ avec $a \in \mathbb{C}$

Suites usuelles

- Variation d'une suite, suite majorée, minorée, bornée.
- suite arithmétique : définition, terme général en fonction de n, somme des termes d'une suite arithmétique.
- suite géométrique : définition, terme général en fonction de n, somme des termes d'une suite géométrique.
- suite arithmético-géométrique : définition, méthode pour déterminer le terme général en fonction de n.

Remarques aux colleurs

- Merci aussi de poser une petite question d'informatique (cf Annexe).
- La notion de nombre complexe est nouvelle pour beaucoup de nos élèves.

Exemples de programmes informatiques

Exercice 1

Réaliser une fonction `DepasseValeur` prenant en paramètre un entier naturel M et renvoyant le plus petit entier naturel n tel que $2^n > M$.

```
def DepasseValeur(M):
    n=0  #initialisation du compteur
    while (2**n <=M):
        n=n+1    #incrementation du compteur
    return n
```

Exercice 2

Créer une fonction `somme` de paramètre n calculant $\sum_{k=0}^{n} k^3$

```
def somme(n):
    S=0
    for k in range(n):
        S = S + k**3
    return S
```

Exercice 3

Créer une fonction récursive `factorielle` qui prend en entrée un entier naturel n et renvoie le résultat de $n!$.

```
def factorielle(n):
    if n==0:
        return 1
    else :
        return n*factorielle(n-1)
```

Exercice 4

Créer une fonction récursive `suite` qui prend en entrée un entier naturel `n` et renvoie la valeur de u_n où (u_n) est la suite définie par :

$$\begin{cases} u_0 = \frac{1}{2} \\ \forall n \in \mathbb{N}, u_{n+1} = 3u_n + 2 \end{cases}$$

```
def suite(n):
    if n==0:
        return 1/2
    else :
        return 3*suite(n-1) +2
```