Semaine n°12 du 18 décembre au 22 décembre

Informatique(Python) : cf exemples en annexe

- Boucle `while` + Compteur.
- Fonction récursive
- Boucle `for` Calcul de somme.
- Parcours de liste. (Attention la modification de liste sera vu la semaine prochaine)

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.
- suite récurrente linéaire d'ordre 2 : définition, équation caractéristique, détermination du terme général en fonction de n.
- Théorème sur les limites d'une suite : théorème de la limite monotone, théorème des gendarmes, théorème de comparaison.

Primitives et intégrales

- Primitives usuelles (cf formulaire) et reconnaissance des composées.
- Intégrale d'une fonction continue sur un segment : définition, expression à l'aide d'une intégrale de l'unique primitive d'une fonction s'annulant en un point.
- Propriété de l'intégrale : linéarité, positivité (démonstration exigible), croissance (démonstration exigible).
- Intégration par parties (démonstration exigible). Application au calcul d'une primitive du logarithme (démonstration exigible).

Remarques aux colleurs

— Merci aussi de poser une petite question d'informatique (cf Annexe).
— Aucun exercice sur les primitives n'a été traité en TD. Pouvez vous vérifier que le formulaire sur les primitives est maitrisé ?

Exemples de programmes informatiques

Exercice 1

Ecrire en Python une fonction `existence` qui prend en entrée une liste `L` et un nombre `element` et renvoie `True` si `element` se trouve dans la liste `L`, `False` sinon.

```
def existence(L,element):
    n=len(L)   # taille de la liste
    for i in range(n):
        if L[i]==element:
            return True
    return False  # si on n' a pas trouvé element après avoir parcouru toute la liste
```

Exercice 2

Ecrire en Python une fonction `MaximumListe` qui prend en entrée une liste `L` et renvoie la plus grande valeur de cette liste

```
def MaximumListe(L):
    n=len(L)  #taille de la liste
    maxi=L[0]  #on considère temporairement que le max est le premier élément
    for i in range(n):
        if L[i]>maxi:
            maxi=L[i]  #on a trouvé une plus grande valeur
    return maxi
```

Exercice 3

Ecrire en Python une fonction `Somme` qui prend en entrée une liste L et renvoie la somme de ses éléments :

```
def Somme(L):
    n=len(L)   #taille de la liste
    S=0  #initialisation de la somme
    for i in range(n):
        S=S+L[i]
    return S
```