Semaine n°7 du 11 novembre au 15 novembre

Informatique(Python) : cf exemples en annexe

- Fonctions : `def`, `return`.
- Instructions conditionnelles `if`, `else`, `elif`. (pas de fonction récursive)
- Module maths et random et variable global/local.
- Script `input`, `print`.
- Boucle `while` + Compteur.

Fonctions usuelles

- Fonctions usuelles (cf formulaire). Pour chaque fonction du formulaire, les domaines de définition, continuité, dérivabilité, dérivée, limites et **graphique** doivent être parfaitement connus :
  - Fonctions affines.
  - Valeur absolue.
  - Partie entière.
  - Fonctions puissances (à exposant entier positif, entier négatif),
  - Racine carrée, racine cubique.
  - Logarithme népérien et logarithme décimal,
  - Exponentielle (base e).
  - exponentielle de base $a \in \mathbb{R}_+^*$.
  - Fonction puissance réelle.
  - Cosinus, sinus et tangente.

Variation de fonction

- Ensemble de définition,
- Compositions de fonctions et recherche du domaine de définition d'une composée.
- périodicité, parité, monotone.
- image d'un ensemble, Théorème des valeurs intermédiaire.
- Lien entre dérivée et monotonie. Théorème de la bijection.
- Majorant ou minorant d'une fonction.
- Extrema : définition et théorème :

  Si f, définie sur un intervalle I, admet un extremum en $x_0 \in I$ et x_0 n'est pas une extrémité de I alors $f'(x_0) = 0$
- Limites : limites usuelles, opération sur les limites, composée de limites, croissance comparée, théorème des gendarmes, théorème de comparaison.
- Asymptotes verticale, horizontale, oblique. ATTENTION : aucune méthode d'étude du comportement asymptotique n'a été vue cette année.

Dérivées

- Nombre dérivé en a : définition, interprétation graphique (coefficient directeur de la tangente), fonction dérivée.
- Définition : fonction de classe $\mathcal{C}^1$.
- Dérivées usuelles.
- Opérations sur les dérivées, compositions, formules de compositions usuelles ($\ln(u)$, e^u, $\sqrt{u}$, u^n).
- Fonctions de deux variables : définition, ensemble de définition, méthode de calcul des dérivées partielles.

Méthodes de calcul : sommes et produits

- Notation $\sum$: définition, linéarité, Chasles, changement d'indice.

Remarques aux colleurs

— Veillez s'il vous plait à ce que la détermination des ensembles de dérivabilité soient bien rédigées et que le raisonnement soit bien compris (voir rédactions en annexe).
— N'hésitez pas à vérifier que les courbes représentatives des fonctions classiques sont connues (point classique et tangente)
— Merci aussi de poser une petite question d'informatique (cf Annexe).

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

On veut créer le programme suivant : l'ordinateur choisit un nombre entier au hasard entre 1 et 100 puis demande à l'utilisateur de rentrer des nombres entiers jusqu'à deviner le nombre choisi par l'ordinateur. A chaque tentative ratée de l'utilisateur, l'ordinateur indique si le nombre proposé est trop grand ou trop petit par rapport au nombre cherché.

```
from random import *
nb=randint(1,100)
tentative=0   #valeur fausse pour pouvoir rentrer dans la boucle
while (tentative != nb):
    tentative=eval(input("donner un nombre entier entre 1 et 100 : "))
    if tentative <nb:
        print("le nombre proposé est trop petit.")
    elif tentative > nb:
        print("le nombre proposé est trop grand.")
    else:
        print("Gagné!")
```

Exercice 3

Creer un script qui demande à l'utilisateur de donner le mot de passe du labo de bio ("cellule"). L'utilisateur a le droit à trois tentatives maximum.

```
mdp="cellule"
proposition=''    #valeur fausse pour pouvoir rentrer dans la boucle
nbessai = 0      #initialisation du compteur
while (proposition != mdp) and (nbessai <3):
    proposition=input("donner le mot de passe : ")
    nbessai = nbessai + 1    #incrementation du compteur
if (proposition ==mdp):
    print("bienvenu au labo de bio")
else:
    print("nombre de tentatives dépassé")
```

Exercice 4

Créer une fonction `entier` qui prend en entrée un entier et qui renvoie True si le nombre est un entier et False sinon

```
from math import floor
def entier(x):
   if (x==floor(x)) :
          return True
    else :
          return False
```

Exercice 5

Créer une fonction `jeu` qui prend en entrée un nombre entier `b` entre 0 et 10 et génère un nombre a au hasard entre 0 et 10. Si `b` et a sont égaux, on renvoie " gagné ", sinon on affiche " perdu ".

```
from random import randint
def jeu(b):
   a=randint(0,10)
   if (a==b) :
        return "gagné"
    else :
        return "perdu"
```

Exercice 6

Créer une fonction python `piecedesequilibre` qui simule l'expérience suivante : On lance une pièce de monnaie qui tombe sur pile avec probabilité 1/4 et qui tombe sur face avec probabilité 3/4.

Première solution :

```
from random import randint
def piecedesequilibre():
     a=randint(1,4)
     if (a==1) :  # 1 chance sur 4 d'obtenir le chiffre 1 quand on choisit un entier
         return "pile"
      else :
         return "face"
```

Deuxième solution :

```
from random import random
def piecedesequilibre():
    a=random() # nombre réel quelconque choisi entre 0 et 1
    if (a<=0.25) : # la proportion de nombres réels a<=0.25 est 0.25
          return "pile"
    else :
          return "face"
```

Exemples de rédaction

Exercice 1

On considère la fonction $f : x \mapsto \sqrt{e^x - 1}$

1. Déterminer le domaine de définition de f.

   solution :

   Soit $x_0 \in \mathbb{R}$,

$$f \text{ est définie en } x_0 \iff \begin{cases} x \mapsto e^x - 1 \text{ est définie en } x_0, \\ e^{x_0} - 1 \geq 0, \qquad\qquad \text{car } x \mapsto \sqrt{x} \text{ est définie sur } \mathbb{R}_+ \end{cases}$$
$$\iff \begin{cases} x_0 \in \mathbb{R}, \quad \text{car } x \mapsto e^x \text{ est définie sur } \mathbb{R}, \\ e^{x_0} \geq 1, \end{cases}$$
$$\iff x_0 \geq 0, \qquad \text{car exponentielle est strictement croissante sur } \mathbb{R}.$$

Le domaine de définition de f est $D_f = [0 \ ; \ +\infty[$.

2. Déterminer le domaine de dérivabilité de g et calculer sa dérivée.
solution :
Soit $x_0 \in \mathbb{R}$,

$$f \text{ est dérivable en } x_0 \text{ si } \begin{cases} x \mapsto e^x - 1 \text{ est dérivable en } x_0, \\ e^{x_0} - 1 > 0, \qquad\qquad \text{car } x \mapsto \sqrt{x} \text{ est dérivable sur } \mathbb{R}_+^* \end{cases}$$
$$\text{si } \begin{cases} x_0 \in \mathbb{R}, \quad \text{car } x \mapsto e^x \text{ est dérivable sur } \mathbb{R}, \\ e^{x_0} > 1, \end{cases}$$
$$\text{si } x_0 > 0, \qquad \text{car exponentielle est strictement croissante sur } \mathbb{R}.$$

Le domaine de dérivabilité de f est $D =]0 \ ; \ +\infty[$.

$$\forall x \in]0 \ ; \ +\infty[, f'(x) = \frac{e^x}{2\sqrt{e^x - 1}}.$$