```
001| ######################################################################
002|      # DEFINITION ET REPRESENTATION DES POLYNÔMES EN LANGAGE PYTHON
003| ######################################################################
004|
005|
006| # Vous trouverez dans ce fichier quelques points de programmation en
Python, qui pourront être utiles pour le PBS9
007|
008| # ILLUSTRATION 1 - Deux manières de définir un polynôme
009| # ILLUSTRATION 2 - Opérations sur les polynômes
010| # ILLUSTRATION 3 - Définir un polynôme par la liste de ses racines
011| # ILLUSTRATION 4 - Tracer la courbe représentative d'un polynôme
012|
013| # Préambule: avant toute chose...
014|
015| import matplotlib.pyplot as plt # Import de la bibliothèque "graphique"
de Python (pour tracer des courbes représentatives)
016| import numpy as np # Import de la bibliothèque numpy (pour les
polynômes)
017|
018|
019| ######### ILLUSTRATION 1 - Deux manières de définir un polynôme
020|
021| # Méthode 1 - Avec la liste des coefficients
022|
023| P = np.poly1d([2,3,4]) # Définit: P = 2X^2 + 3X + 4
024| # Dans le shell, on obtient alors:
025| #>>> P
026| #   poly1d([2, 3, 4])
027|
028|
029| # Méthode 2 - Avec la liste de ses racines
030|
031| Q = np.poly1d([2,3,4],True) # Définit: Q = (X-2)(X-3)(X-4)
032| # Dans le shell, on obtient alors:
033| #>>> Q
034| #   poly1d([  1.,  -9.,  26., -24.])
035| # Ce qui signifie que Q = X^3 - 9X^2 + 26X - 24 ("la machine retourne
la forme développée de Q")
036|
037| ######### ILLUSTRATION 2 - Opérations sur les polynômes (syntaxe)
038|
039| # Somme de deux polynômes P et Q:  P + Q
040| # Multiplication d'un polynôme P par un scalaire (3 par exemple): 3 * P
041| # Evaluation d'un polynôme P en un réel (2 par exemple): P(2)
042|
043| ######### ILLUSTRATION 3 - Définir un polynôme par la liste de ses
racines
044|
045| # Généralisation de la méthode 2
046| # On construit une liste, par exemple la liste des carrés des entiers
entre 0 et 6
047|
048| liste_rac = [k**2 for k in range(7)] # Ainsi: liste_rac = [0, 1, 4, 9,
16, 25, 36]
049|
050| # On définit le polynôme (unitaire) R ayant pour racines les éléments
de liste_rac en posant:
051|
052| R = np.poly1d(liste_rac,True)
```

```
053|
054| ######### ILLUSTRATION 4 - Tracer la courbe représentative d'un
polynôme
055|
056| # Dans cet exemple, on se propose de tracer la courbe représentative de
S = X(X-1)(X-2)(X-3)(X-4)
057| # sur l'intervalle [-1,5] par exemple
058|
059|
060| # 1/ Définition de S
061|
062| liste_racS = [k for k in range(5)]
063| S = np.poly1d(liste_racS,True)
064|
065|
066| # 2/ Préparation du graphique
067|
068| NPTS = 1000 # Nombre de points sur la "courbe" représentative
069| xmin = -1 # Comme son nom le suggère assez bien...
070| xmax = 5 #
071| h = (xmax-xmin)/NPTS # Définition du pas
072| plt.clf(); # Initialisation de la fenêtre graphique
073| axes = plt.gca() # Style des axes
074| plt.grid(True) # Grille de fond
075| axes.set_xlim(xmin, xmax) # Valeurs limites de l'abscisse
076| axes.set_ylim(-4, 4) # Valeurs limites de l'ordonnée
077|
078|
079| # 3/ Création de la liste des abscisses (subdivision de l'intervalle
[xmin, xmax])
080|
081| x = xmin
082| LABS = [x]
083| for k in range(NPTS):
084|     x = x+h
085|     LABS = LABS + [x]
086|
087|
088| # 4/ Création de la liste des ordonnées (images de S aux points de la
subdivision)
089|
090| LORD = []
091| for k in range(len(LABS)):
092|     LORD = LORD + [ S(LABS[k]) ]
093|
094|
095| # 5/ Affichage du graphique
096|
097| plt.plot(LABS, LORD)
098| plt.show();
099|
100|
101| ############ FIN ############
```