Algorithmique avec Python Terminale spécialité maths

1. Des sites pour apprendre:

Le site d'Yvan Monka : https://www.maths-et-tiques.fr/index.php/tp-info/tutoriels-calculatrices

Python au lycée : http://python.lycee.free.fr/

Site académie de Créteil : https://math93.com/index.php/divers/enseignants/python-au-lycee

France IOI (exercices) : http://www.france-ioi.org/algo/chapters.php

Un site très complet et bien documenté (un cours bien fait, mais pas d'exercices) :
http://apprendre-python.com/ (https://python.doctor/)

2. Des environnement de travail pour PYTHON :

- EduPython est une distribution clé en main, complète et portable pour programmer avec vos élèves sous un environnement Python 3.
  Un inconvénient : Elle ne fonctionne que sous Windows.
- Anaconda présente elle aussi de nombreux avantages. Elle comporte les modules nécessaires au lycée, un éditeur performant : Spyder et a l'avantage d'être multi-plateforme : Windows, Linux, MAC OS. Un peu lourde toutefois.
- Winpython est une autre distribution portable entièrement libre, mais spécifique à Windows. Elle utilise l'éditeur Spyder et comporte les librairies nécessaires au lycée.
- Pyzo est également un environnement très agréable. Accompagné de la distribution miniconda, il offre un éditeur performant et les modules nécessaires au lycée. Il fonctionne sous tous les systèmes d'exploitation.
- Thonny : https://thonny.org/ est un IDE (environnement de développement) minimaliste qui permet d'apprendre le Python. Conçu pour les débutants, il intègre son propre interpréteur Python 3.6, et offre des fonctionnalités plutôt sympas quand on est dans un processus d'apprentissage.
- Si vous souhaitez un environnement plus léger, avec les fonctionnalités de base, mais parfaitement fonctionnel, vous pouvez installer Python 3 avec l'éditeur IDLE livré avec et disponible pour tous les systèmes d'exploitation. Dans ce cas, il vous faudra installer manuellement les librairies supplémentaires.
- Vous pouvez également utiliser Geany très léger et multi-langages, mais qui nécessitera lui aussi une installation manuelle des modules complémentaires.

ALGORITHMIQUE TERMINALE S

1. Quelques instructions en Python et T.I. :

	Instruction	Python	T.I.
Saisie par l'utilisateur	Saisir la valeur de l'entier n	n = int(input("n = "))	Prompt N
	Saisir la valeur du réel x	x = float(input("x = "))	Prompt X
Notation des puissances	a prend la valeur 2^5	A = 2**5	$2^5 \rightarrow$ A
Instructions conditionnelles	Si ($delta == 0$) alors $x \leftarrow -b/(2a)$	If (delta == 0) : x = -b/(2*a)	:If (D = 0) :Then -B/(2*A) → X :End

	Si $a > 2$ alors $a \leftarrow 2a$ sinon $a \leftarrow 3a$	If (a > 2) :
		  a = 2*a
		else :
		  a = 3*a
			:If A > 2
			:Then 2*A → A
			:Else 3*A → A
			:End

Boucle Pour	Pour i variant de 0 à $n-1$ faire	for i in range(n):	:For(I,0,N-1)
	Pour i variant de 1 à n faire	for i in range(1,n+1):	:For(I,1,N)

Boucle Tant Que	Tant que $u < 100$ faire
	  u prend la valeur $2u+1$
		while (u > 100) :
		  u = 2*u+1
			:While (U > 100)
			:2*U+1 → U
			:End

2. Exemples :

a) Algorithme de la résolution de l'équation du second degré :

Algorithme	Python	T.I.

Saisir la valeur du réel a
Saisir la valeur du réel b
Saisir la valeur du réel c
delta = $b^2 - 4ac$
Si (delta < 0) alors afficher : « pas de solution »
Si (delta $== 0$) alors afficher : « une solution », $-b/(2a)$
Si (delta > 0) alors afficher : « deux solutions »,
$(-b - \sqrt{\text{delta}})/(2a)$, $(-b + \sqrt{\text{delta}})/(2a)$
	a = float(input("a="))
	b = float(input("b="))
	c = float(input("c="))
	d = b**2-4*a*c
	if (d < 0) :
	  print("pas de solutions")
	if (d == 0) :
	  print("une solution = ",
	  -b/(2*a))
	if (d>0):
	  print("2 solutions")
	  print((-b-sqrt(d))/(2*a))
	  print((-b+sqrt(d))/(2*a))
		:Prompt A, B, C
		:B² – 4*A*C → D
		:If D < 0
		:Then Disp « PAS DE SOL »
		:End
		:If D = 0
		:Then Disp « UNE SOL »,
		-B/(2*A)
		:End
		:If D > 0
		:Then Disp « DEUX SOL »,
		(-B - √D)/(2*A), (-B + √D)/(2*A)
		:End

b) Calcul des premiers termes de la suite : u_0 donné et $u_{n+1} = \dfrac{2u_n}{1+u_n} + 1$.

Algorithme	Python	Variante Python

u prend la valeur u_0
n prend la valeur nombre de termes
Pour i variant de 0 à $n-1$ faire
  u prend la valeur $2u/(1+u)+1$
  Afficher u
	u = float(input("u0 = "))
	n = int(input("nombre de termes :"))
	
	for i in range(n):
	  u = 2*u/(1+u)+1
	  print(u)
		def f(u):
		  return 2*u/(1+u)+1
		
		u = float(input("u0 = "))
		n = int(input("nombre de termes :"))
		for i in range(n):
		  u = f(u)
		  print(u)

3. a) Que fait l'algorithme ci-contre ?
b) L'écrire en langage Python.
c) Déterminer la valeur de sortie avec $n = 10$.

$u \leftarrow 2$; s $\leftarrow u$
Saisir la valeur de n
Pour i allant de 1 à n faire
  $u \leftarrow 2u - 1$
  $s \leftarrow s + u$
Afficher s