TI83CE – Python – N22P66 « résolution de l'équation du second degré ».

Enoncé

Livre Barbazo Math Première Générale

n°21 et n°22 p66

21 Résoudre dans $\mathbb{R}$ les équations suivantes.

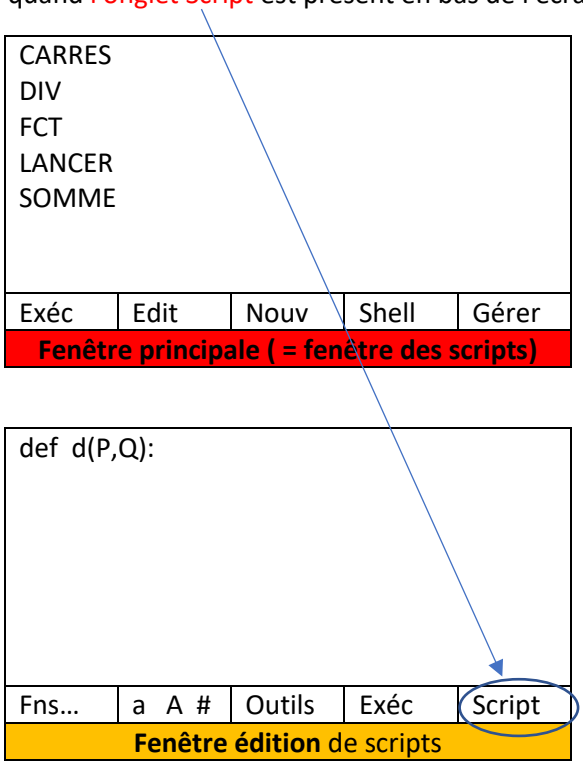
1. $-2x^2 + x - 1 = 0$
2. $2x^2 - 2x - 1 = 0$
3. $5x^2 - 2x + 1 = 0$
4. $2x^2 + 4 = -6x$
5. $x(2x - 1) = 1$
6. $x^2 = -5x - 1$
7. $-x + 3x^2 - 1 = 0$
8. $x(8 - x) + 1 = 0$
9. $2x^2 + 6x + \frac{9}{2} = 0$
10. $x^2 + 2\sqrt{3}x + 3 = 0$
11. $-3x^2 + x = -\frac{1}{4}$
12. $2x(5 + 2x) = 9 - 2x$

22 ALGO PYTHON

1. Écrire un algorithme en langage naturel qui renvoie les solutions de l'équation $ax^2 + bx + c = 0$, avec a, b et c réels et $a \neq 0$, lorsqu'elles existent.
2. Le traduire en langage Python.
3. Le programmer et le tester sur les équations de l'exercice précédent.

TI83CE – Python – N22P66 « résolution de l'équation du second degré ».

La **fenêtre principale est la fenêtre des scripts** (c'est celle sur laquelle s'ouvre Python). Pour l'atteindre à tout moment, appuyer sur la touche F5 quand l'onglet Script est présent en bas de l'écran.

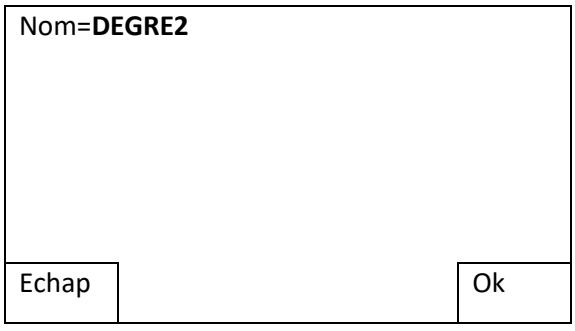


n22 p66 :

Question 2

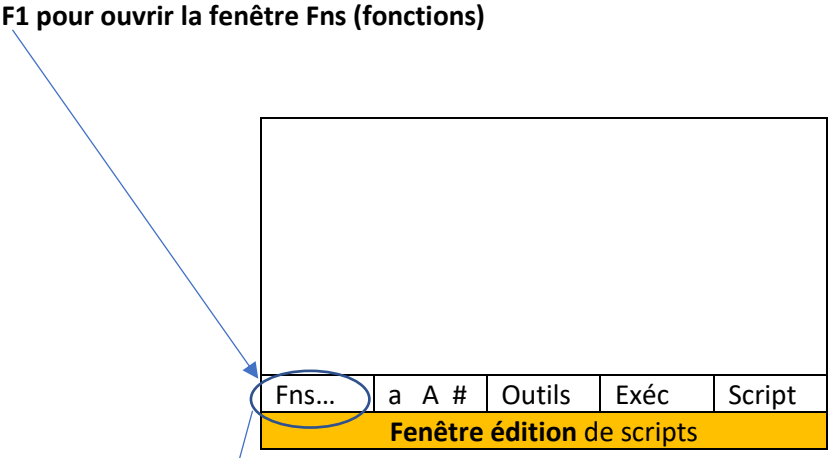
Depuis la fenêtre principale des scripts, appuyez sur la touche F3 "Nouv"

On entre un nom de script **DEGRE2** (En majuscules seulement, ou chiffres, le premier caractère étant une lettre. Maximum 8 caractères) puis OK.

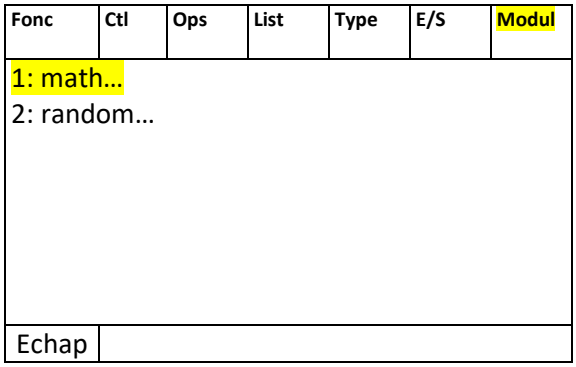


La calculatrice passe automatiquement en fenêtre **édition de scripts**

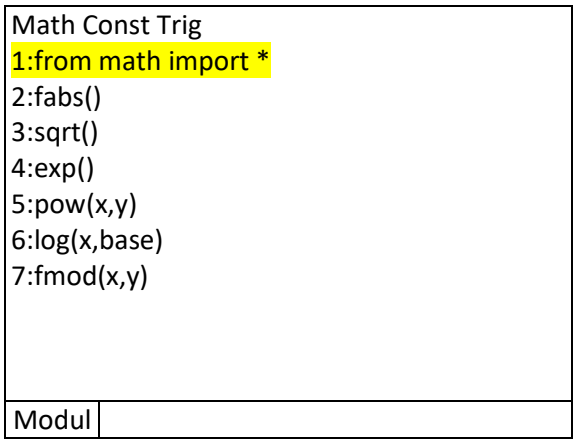
Appuyer sur F1 pour ouvrir la fenêtre Fns (fonctions)



Avec les flèches de direction, allez sur **Modul** pour sélectionner l'importation du module math nécessaire au programme. Sélectionnez **1 :math…** puis appuyez sur la touche Entrer.



On obtient :

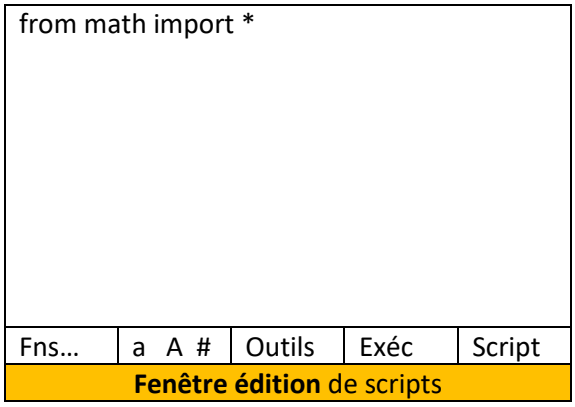


Choisissez **1:from math import *** puis appuyez sur la touche Entrer.

l'étoile * veut dire que cela importe toutes les fonctions du module math, dont la racine carrée que nous allons utiliser et qui est sqrt()

Appuyez sur la touche Entrer.

Vous devriez obtenir l'écran :



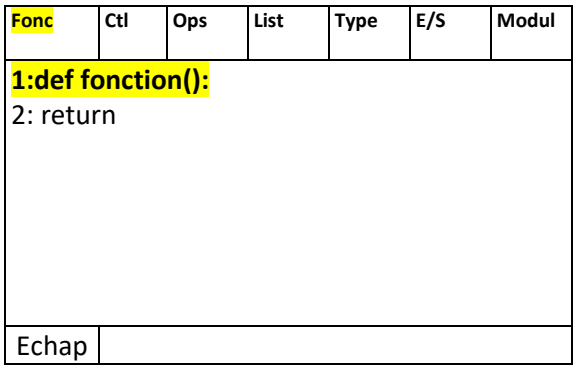
Remarque : A ce niveau, il est possible de n'importer que la fonction que nous allons utiliser.

Ce qui donnerait **from math import sqrt** au lieu de from random import *

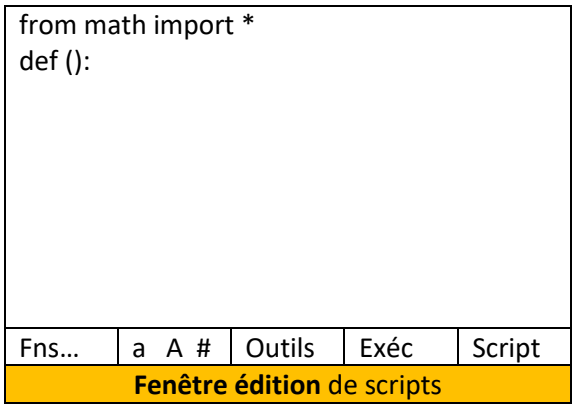
Mais, comme on aura besoin plus loin d'une autre fonction du module math, il vaut mieux garder l'étoile.

Appuyez sur **Fns**

La fenêtre Fns (fonctions) s'ouvre sur l'onglet Fonc déjà activé et sur l'option 1:def fonction(): déjà activée.



Appuyez sur la touche Entrer pour valider ce choix. On obtient :



Le curseur clignote sur la première parenthèse de def () :

Donc, tout ce qui est saisi va se placer juste *avant* la première parenthèse.

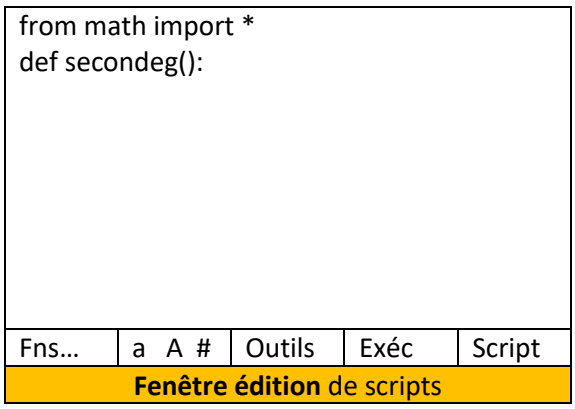
Mettez le clavier en mode alphabétique en appuyant sur 2nd alpha.

On voit que le clavier est en mode alphabétique minuscules par l'apparition de la lettre **a** en haut à droite de l'écran.

Saisissez le nom de la première fonction **secondeg** tout attaché – ne pas mettre d'espace dans un nom de fonction. Cela provoquerait une erreur.

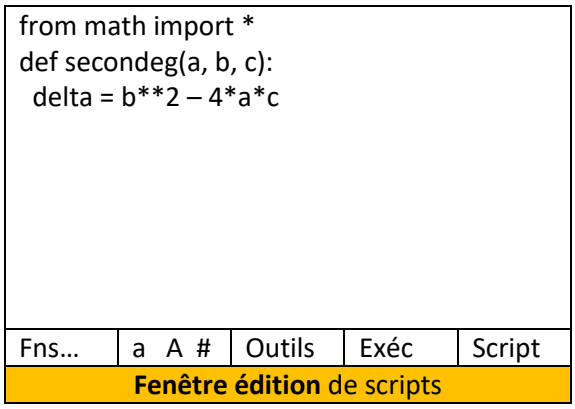
Placez le curseur avec les flèches de direction *à la fin de la ligne* def secondeg() : *après les deux points* et appuyez sur la touche Entrer.

On a alors l'écran :



On remarque que l'indentation Python (qui est de deux espaces sur les TI83 Python) se fait automatiquement.

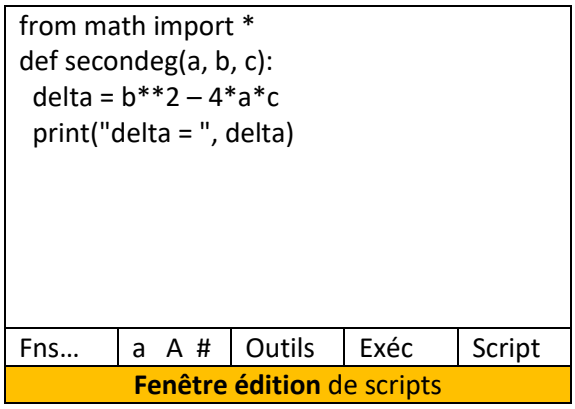
Saisissez le code suivant. S'il se produit une erreur, la **touche suppr** permet de supprimer le caractère *avant* le curseur. Vous devez obtenir le code Python suivant :



Remarques :

- Une étoile est le symbole de la multiplication, deux étoiles est la mise à la puissance.
- Pour le signe = , le plus simple est de faire **2nd test**.
- N'oubliez pas de faire 2nd alpha pour sortir du mode alphabétique et pouvoir saisir les signes d'opérations et les chiffres.

Continuez jusqu'à obtenir :



Remarque : Pour print(), allez dans **Fns** puis **E/S** (situé en haut de la fenêtre) puis choisissez print(): puis appuyez sur la touche Entrer.

Continuez jusqu'à obtenir :

```
from math import *
def secondeg(a, b, c):
  delta = b**2 – 4*a*c
  print("delta = ", delta)
  if delta < 0:
    print("Pas de solution")
  elif :

  else:
```

Fns…	a A #	Outils	Exéc	Script

Fenêtre édition de scripts

Remarques :

Pour if .. elif .. else .., allez dans **Fns** puis **Ctl** (les instructions de contrôle) puis choisissez if .. elif .. else puis appuyez sur la touche Entrer. **elif** est un autre if après un premier if. On peut mettre plusieurs elif, tous au même niveau d'indentation que le premier if. **else** se met éventuellement à la fin pour traiter tous les autres cas qui n'ont pas été vus dans les if et elif placés avant.

Continuez jusqu'à obtenir

```
from math import *
def secondeg(a, b, c):
  delta = b**2 – 4*a*c
  print("delta = ", delta)
  if delta < 0:
    print("Pas de solution")
  elif delta == 0 :
    x0 = -b/(2*a)
    print("x0 = ", x0)
  else:
    x1 = (-b + sqrt(delta))/(2*a
      )
```

Fns…	a A #	Outils	Exéc	Script

Fenêtre édition de scripts

Rappel : le signe d'affectation = et le test d'égalité == se trouvent dans 2nd test

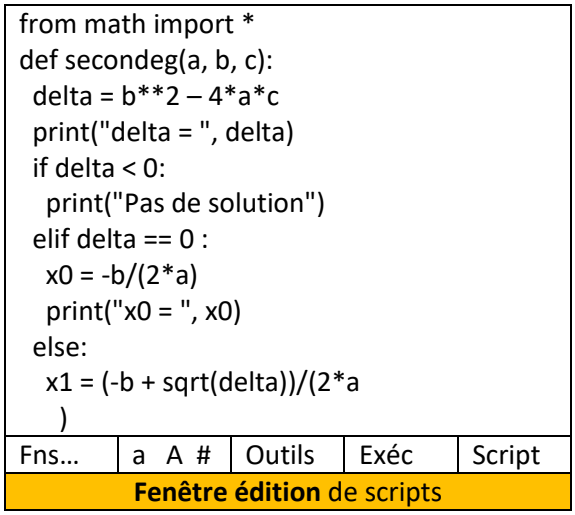
On va utiliser un copier-coller pour obtenir :

```
  if delta < 0:
    print("Pas de solution")
  elif delta == 0 :
    x0 = -b/(2*a)
    print("x0 = ", x0)
  else:
    x1 = (-b + sqrt(delta))/(2*a
      )
    x2 = (-b - sqrt(delta))/(2*a
      )
```

Fns…	a A #	Outils	Exéc	Script

Fenêtre édition de scripts

Pour cela depuis l'écran :



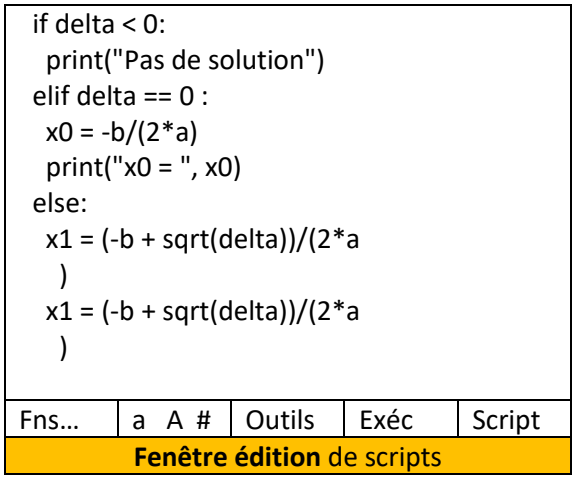
Placez le curseur quelque part sur la ligne x1 = (-b + sqrt(delta))/(2*a

Allez dans l'onglet Outils (touche F3) sélectionnez 6 :Copier Ligne

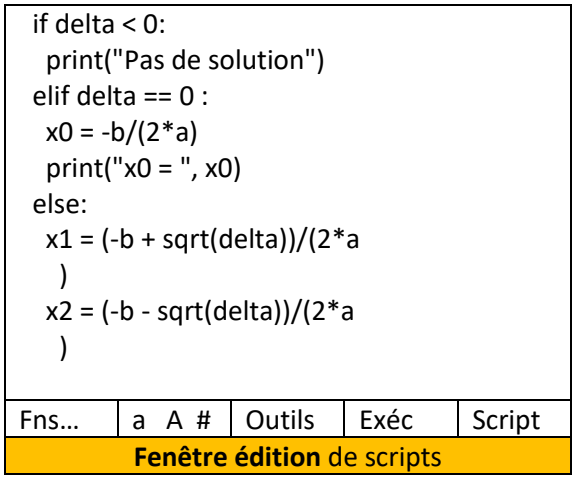
Placez le curseur juste *après* la parenthèse fermante de la dernière ligne :

Allez dans l'onglet Outils (touche F3) sélectionnez 7 :Coller Ligne

Vous devez obtenir



Il suffit de changer le + en – et le 1 en 2 pour avoir :



Remarque : Avec « Outils », on peut supprimer une ligne directement.

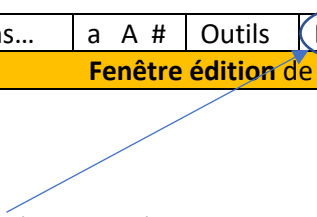
Il faut placer le curseur sur la ligne à supprimer, aller dans **Outils**, choisir **5:Couper Ligne**

Terminez le code :

```
elif delta == 0 :
  x0 = -b/(2*a)
  print("x0 = ", x0)
else:
  x1 = (-b + sqrt(delta))/(2*a
    )
  x2 = (-b - sqrt(delta))/(2*a
    )
  print("x1 = ", x1)
  print("x2 = ", x2)
```

Fns…	a A #	Outils	Exéc	Script

Fenêtre édition de scripts



Pour exécuter ce script, allez dans Exéc (touche F4)

La calculatrice passe en mode shell (ou mode console) ce qui est visible parce que les lignes commencent par 3

chevrons >>>

Le message

```
>>> # L'exécution de DEGRE2
>>> from DEGRE2 import *
```

Cela indique que toutes les fonctions (en l'occurrence il n'y en a qu'une, c'est **secondeg()** ont été importées du script DEGRE2 dans la mémoire de travail de la calculatrice.

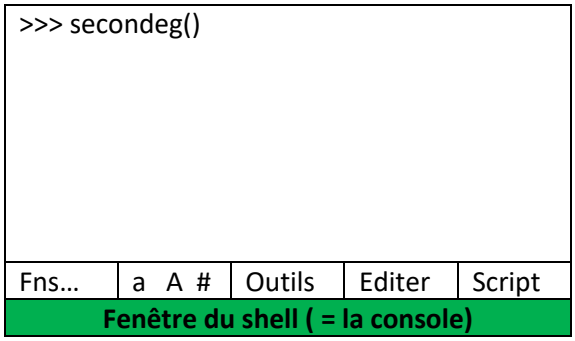
Remarque : si vous voulez nettoyer la console, allez dans **Outils** et choisissez **5:Effacer l'écran**

Appuyer sur la touche **var**

choisissez la fonction **secondeg()**

puis OK (touche F5)

L'écran suivant apparait :



Test de la fonction secondeg() avec les exemples du n°21 p66

Pour tester la fonction, il faut lui indiquer les valeurs des 3 paramètres a, b, c. Par exemple >>> secondeg(-2,1,-1)

1	$-2x^2+x-1=0$ $\Delta=-7$ Pas de solution	2	$2x^2-2x-1=0$ $\Delta=12$ $x_1=\frac{1-\sqrt{3}}{2}\approx-0{,}37\quad x_2=\frac{1+\sqrt{3}}{2}\approx 1{,}37$
3	$5x^2-2x+1=0$ $\Delta=-16$ Pas de solution	4	$2x^2+6x+4=0$ $\Delta=4$ $x_1=-2\quad;\quad x_2=-1$
5	$2x^2-x-1=0$ $\Delta=9$ $x_1=-\frac{1}{2}\quad;\quad x_2=1$	6	$x^2+5x+1=0$ $\Delta=21$ $x_1=\frac{-5-\sqrt{21}}{2}\approx-4{,}791\quad x_2=\frac{-5+\sqrt{21}}{2}\approx-0{,}21$
7	$3x^2-x-1=0$ $\Delta=13$ $x_1=\frac{1-\sqrt{13}}{6}\approx-0{,}43\quad x_2=\frac{1+\sqrt{13}}{6}\approx 0{,}77$	8	$-x^2+8x+1=0$ $\Delta=68$ $x_1=\frac{8+2\sqrt{17}}{2}\approx 8{,}123\quad x_2=\frac{8-2\sqrt{17}}{2}\approx-0{,}123$
9	$2x^2+6x+\frac{9}{2}=0$ $\Delta=0$ $x_0=-\frac{1}{2}$	10	$x^2+2\sqrt{3}x+3=0$ $\Delta=0$ $x_0=-\sqrt{3}$
11	$-3x^2+x+\frac{1}{4}=0$ $\Delta=4$ $x_1=\frac{1}{2}\quad;\quad x_2=-\frac{1}{6}\approx-0{,}17$	12	$4x^2+12x-9=0$ $\Delta=288$ $x_1=\frac{-3-3\sqrt{2}}{2}\approx-3{,}621\quad x_2=\frac{-3+3\sqrt{2}}{2}\approx 0{,}621$

Remarques :

Pour relancer l'exécution de la fonction, il suffit d'appuyer sur la touche de direction "vers le haut"

Pour ajouter des indentations sur une ligne, il suffit de se placer à l'endroit où on veut ajouter des indentations (décalages) et choisir Indent ▸ dans le menu **Outils**.

Problème du cas n°10

Si on exécute

>>> secondeg(1,2*sqrt(3),3)

on obtient

delta = -1.77e-15

Pas de solution

L'origine de l'erreur (on devrait trouver delta = 0) est que les nombres flottants (les nombres à virgule flottante) sur les machines ont une précision limitée (ici de l'ordre de 1.10^{-15})

Donc pour comparer deux nombres flottants (par exemple ici **1.77e-15** et **0**) on ne teste pas l'égalité entre les deux nombres. On vérifie s'ils sont suffisamment proches. Dans notre cas, "suffisamment" signifie "plus gros que l'erreur" qui est de l'ordre de 1.10^{-15}. Donc on peut décider que tout écart inférieur à 1.10^{-14} sera considéré comme étant nul. Ainsi, delta = -1.77e-15 sera traité dans l'algorithme, comme si Python trouvait delta = -0

On contrôle si la valeur absolue de la différence entre delta et 0 est inférieure à 1.10^{-14}. La valeur absolue d'une différence est la distance entre les deux nombres.

La fonction valeur absolue en Python sur la TI83 est la fonction **fabs**. Elle se trouve dans le module math.

Modifiez le code de la façon suivante :

```
elif fabs(delta) < 1e-14 :
  x0 = -b/(2*a)
  print("x0 = ", x0)
else:
  x1 = (-b + sqrt(delta))/(2*a
    )
  x2 = (-b - sqrt(delta))/(2*a
    )
  print("x1 = ", x1)
  print("x2 = ", x0)
```

Fns…	a A #	Outils	Exéc	Script

Fenêtre édition de scripts

Testez à nouveau le cas n°10.

On constate que le problème n'a pas disparu.

Cette fois, l'origine du défaut est *qu'on teste en premier si delta < 0.*

Comme Python trouve encore delta = -1.77e-15 alors dans le premier if, l'instruction print("Pas de solution") est exécutée.

Il suffit changer l'ordre : on met le test if **fabs(delta) < 1e-14: en premier**

Dans ce cas, si Python trouve delta = -1.77e-15, c'est bien l'instruction de calculer x0 qui sera exécutée et non plus l'instruction d'afficher "Pas de solution".

Modifiez votre code de façon à obtenir :

```
from math import *
def secondeg(a, b, c):
  delta = b**2 – 4*a*c
  print("delta = ", delta)
  if fabs(delta) < 1e-14 :
    x0 = -b/(2*a)
    print("x0 = ", x0)
  elif delta < 0 :
    print("Pas de solution")
  else:
    x1 = (-b + sqrt(delta))/(2*a
      )
```

Fns…	a A #	Outils	Exéc	Script

Fenêtre édition de scripts

Vérifiez le bon fonctionnement de cette fonction Python pour les exemples du n°21 p66, y compris le cas n°10.

Vous devriez voir pour le cas n°10 s'afficher :

delta = -1.77e-15

x0 = -1.732

A vous de savoir qu'il s'agit de valeurs approchées et qu'une quantité de l'ordre de 10^{-15} lorsqu'elle est affichée par la machine est à interpréter comme étant 0. De même la valeur de la racine x_0 est une valeur approchée de $-\sqrt{3}$.

Remarques :

On peut utiliser sur la calculatrice l'**application PlySmlt2 1:RACINES D'UN POLYNOME**

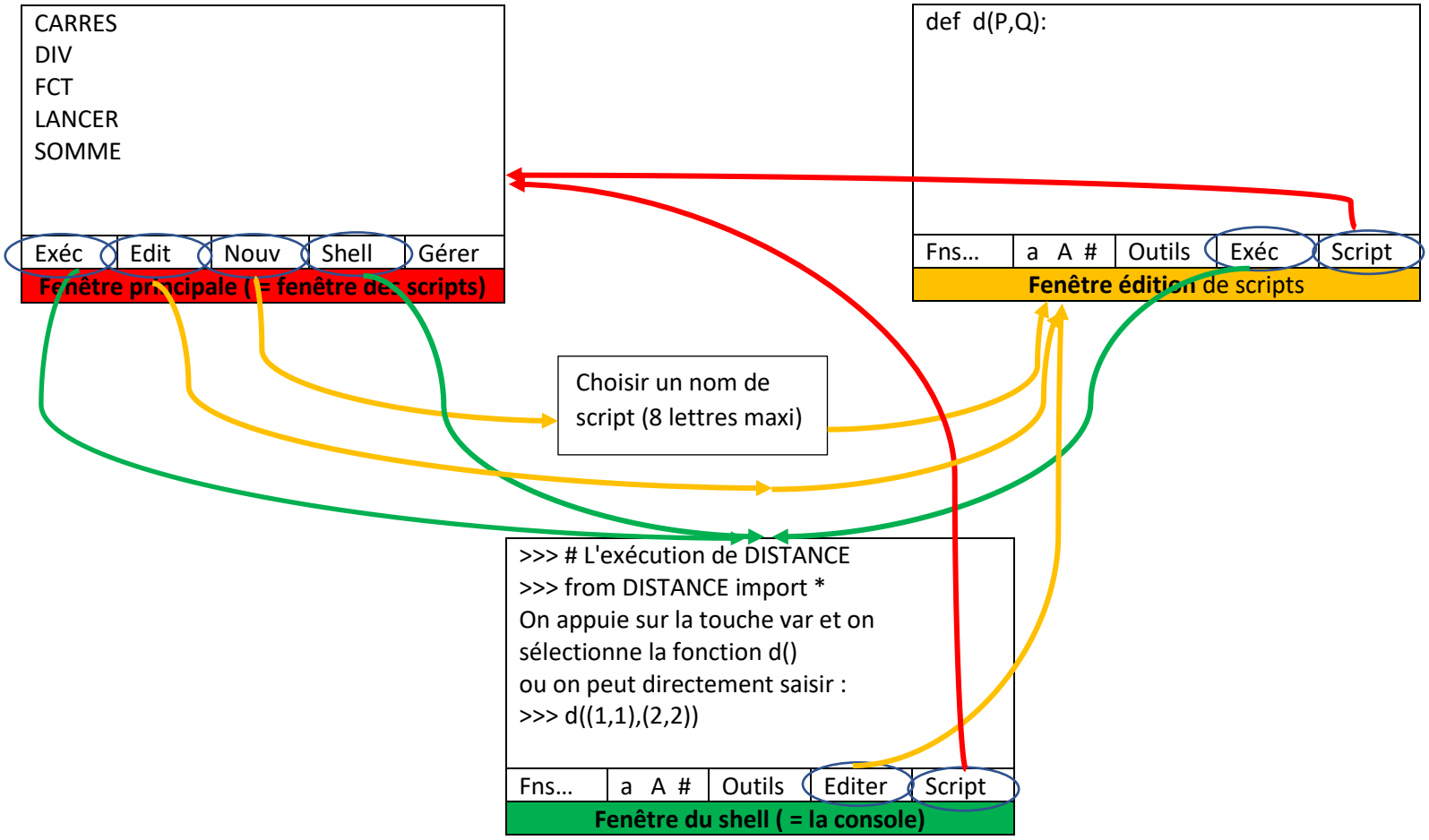
Cette application ne donne pas le discriminant, mais elle donne souvent les valeurs exactes des racines.

Elle est donc complémentaire de votre **script Python DEGRE2**.

Enfin, dans certains cas, l'application PlySmlt2 1:RACINES D'UN POLYNOME ne trouve pas non plus les valeurs exactes. Il vous faudra donc toujours connaitre les formules du discriminant et des racines !

D'autant plus, que parfois, on est amené à les utiliser pour du calcul littéral, c'est-à-dire avec des lettres et non des valeurs numériques pour les coefficients a, b, c.

La page suivante est une feuille de résumé des *trois types de fenêtres Python* présentes sur la calculatrice (scripts, édition, shell) et donne les circuits pour passer d'une fenêtre à l'autre.


CARRES
DIV
FCT
LANCER
SOMME
Exéc
Edit
Nouv
Shell
Gérer
Fenêtre principale (= fenêtre des scripts)
def d(P,Q):
Fns…
a A #
Outils
Exéc
Script
Fenêtre édition de scripts
Choisir un nom de script (8 lettres maxi)
>>> # L'exécution de DISTANCE
>>> from DISTANCE import *
On appuie sur la touche var et on sélectionne la fonction d()
ou on peut directement saisir :
>>> d((1,1),(2,2))
Fns…
a A #
Outils
Editer
Script
Fenêtre du shell (= la console)