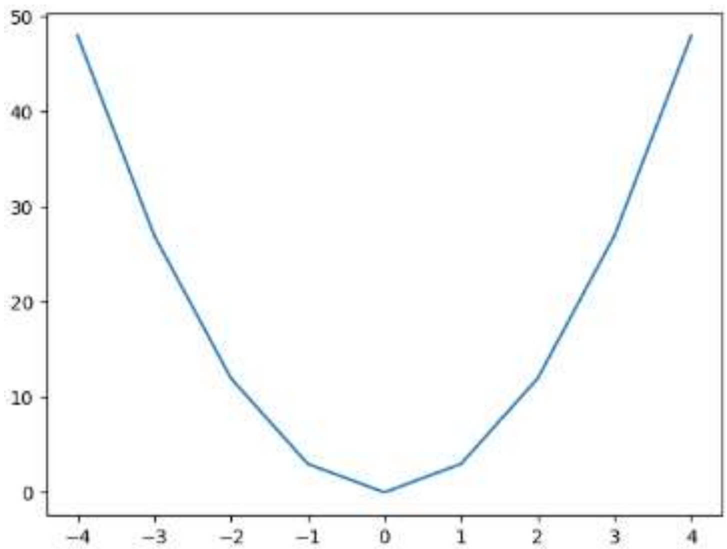
<u>Situation 1</u>

On considère la fonction f définie par $f(x)=x^3$. On souhaite écrire un algorithme qui construit point par point la dérivée de la fonction f, c'est-à-dire la fonction f' qui à tout x associe le nombre dérivé de la fonction f en x. Pour cela on décide d'approximer les valeurs de $f'(x)$ par le taux de variation $\dfrac{f(x+h)-f(x)}{h}$ calculé pour une valeur de h « assez petite ».

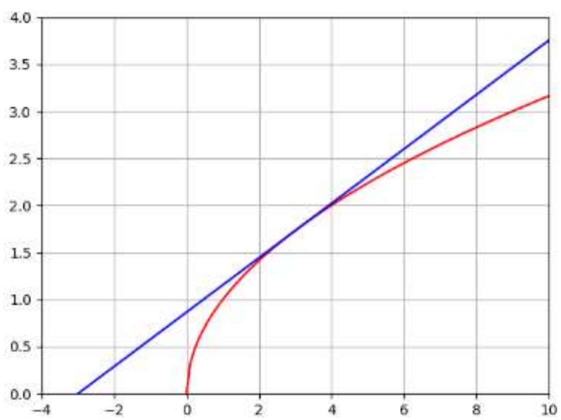
```python
import matplotlib.pyplot as plt

def f(x):
  return(...)

def variation(f,a,h):
  return(...)

def derive(f,xmin,xmax,p,h):
  a=...
  abs=[]
  ord=[]
  while ...:
    abs.append(...)
    ord.append(...)
    a=...
  plt.plot(...)
  plt.show()

derive(f,-4,4,1,1e-10)
```



Compléter le script proposé ci-dessus afin qu'il affiche l'allure de la courbe représentative de la dérivée de la fonction cube sur l'intervalle $[-4;4]$ avec un pas $p=1$ et une valeur de $h=10^{-10}$.
Modifier le programme précédent afin d'obtenir l'allure de la courbe de la fonction dérivée d'autres fonctions de votre choix (on pourra tester sur $g(x)=x^3-7x$ puis sur $h(x)=\sqrt{4x+1}$).

<u>Situation 2</u>

On considère la fonction racine carrée. On cherche à étudier la position l'abscisse du point d'intersection entre l'axe des abscisses et la tangente à la courbe de la fonction f au point d'abscisse a. Conjecturer puis démontrer…




```python
import matplotlib.pyplot as plt
import numpy as np
from math import sqrt

def f(x):
  return(sqrt(x))

def fprime(x):
  return(1/(2*sqrt(x)))

def traceracine(xmin,xmax):
  abs=np.linspace(xmin,xmax,100)
  ord=np.sqrt(abs)
  plt.plot(abs,ord,"r-")

def tracetangente(f,a):
  abs=np.linspace(-10,10,100)
  ord=fprime(a)*(abs-a)+f(a)
  plt.plot(abs,ord,"b-")

plt.axis([-4,10,0,4])
plt.grid()
traceracine(0,10)
tracetangente(f,3)
plt.show()
```

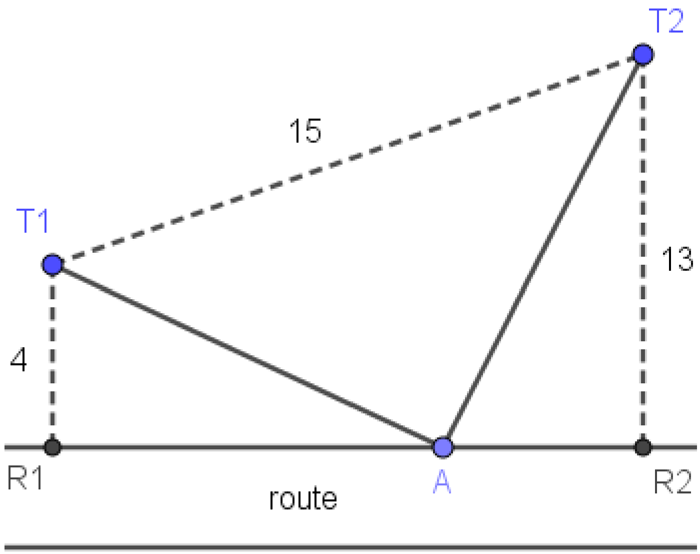
Situation 3

Dans une ville on souhaite relier deux tours T1 et T2 à la fibre. Les distances sont données en centaines de mètres. On prévoit que les câbles suivent une structure rectiligne. Pour différentes raisons, les câbles doivent partir d'un relai A situé le long de la route. On souhaite déterminer la position du point A pour laquelle la longueur totale des deux structures rectilignes soit minimale.

On note $x = AR_1$. Déterminer en fonction de x la distance $d(x) = AT_1 + AT_2$.

On propose ci-dessous un algorithme de recherche des coordonnées du minimum d'une fonction sur un intervalle donné à une précision donnée. Compléter cet algorithme puis l'inclure dans un programme permettant de déterminer au mètre près la position du point A sur la route.

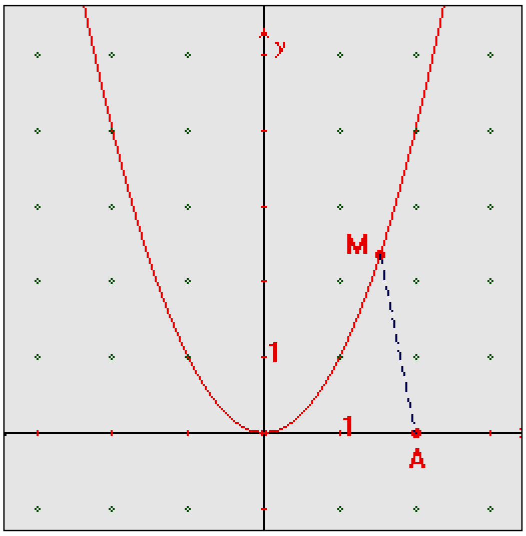
```
def minfonction(f,xmin,xmax,p):
    a=...
    l=[]
    while a<=...:
        l.append(...)
        a=...
    i=l.index(min(l))
    return(...,...)
```



Situation 4

Dans un repère orthonormal, on note P la parabole d'équation $y = x^2$ et A le point de coordonnées $(2;0)$. M est un point quelconque de la parabole P d'abscisse x.

A l'aide d'une figure dynamique conjecturer quelle est la valeur de x pour laquelle la distance AM est minimale. Montrer que $AM^2 = x^4 + x^2 - 4x + 4$. Calculer la dérivée de la fonction f définie par $f(x) = x^4 + x^2 - 4x + 4$.



Nous savons que le minimum d'une fonction est à déterminer parmi les annulations de la dérivée de cette fonction. Compléter le script proposé ci-contre puis l'intégrer dans un programme permettant de déterminer un encadrement d'amplitude inférieure au millième de l'abscisse du point M pour laquelle la distance AM dont parle le problème ci-dessus est minimale.

```
def zero(f,xmin,xmax,p):
    u=...
    v=...
    while ...:
        m=(u+v)/2
        if ...<0:
            v=...
        else:
            u=...
    return(u,v)
```