

TP13 En Python

Représentation graphique : Fonction Carré

Allumer l'ordinateur et connectez-vous en utilisant votre login et votre mot de passe puis lancer « Python en ligne : <https://www.codabrainy.com/python-compiler/> »

Activité

Ce travail porte sur l'étude et la représentation graphique de la fonction $f(x) = x^2 + 2x + 1$ à l'aide de Python. Après analyse, la fonction a été réécrite sous la forme $(x + 1)^2$, ce qui montre qu'il s'agit d'une parabole dont le sommet est au point $(-1 ; 0)$. Un programme a ensuite été développé pour calculer les valeurs de la fonction et tracer sa courbe avec une bibliothèque graphique. Ce travail met en évidence le lien entre mathématiques et informatique, en utilisant la programmation comme outil de visualisation et de compréhension des fonctions.

Algorithme N°1

```
#Algorithme N°1
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return x**2 + 2*x + 1

# Domaine de valeurs
x = np.linspace(-10, 10, 400)
y = f(x)

# Tracé
plt.plot(x, y, label="f(x) = x² + 2x + 1")
plt.axhline(0)
plt.axvline(0)
plt.grid(True)
plt.legend()
plt.title("Représentation graphique de f(x)")
plt.show()
```

Algorithme N°2

On considère la fonction $g(x) = -f(x) = \dots\dots\dots$

En appliquant l'**algorithme 1**, déterminer l'**algorithme 2** permettant de représenter la fonction g .

Algorithme N°3

On considère la fonction $h(x) = -f(x) + 14 = \dots\dots\dots$

En appliquant l'**algorithme 1**, déterminer l'**algorithme 3** permettant de représenter la fonction h .

Algorithme N°4

On considère la fonction $k(x) = f(x) - 14 = \dots\dots\dots$

En appliquant l'**algorithme 1**, déterminer l'**algorithme 4** permettant de représenter la fonction k .

```
#Algorithme N°1
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return x**2 + 2*x + 1

# Domaine de valeurs
x = np.linspace(-5, 5, 400)
y = f(x)

# Tracé
plt.plot(x, y, label="f(x) = -x² - 2x - 1")
plt.axhline(0)
plt.axvline(0)
plt.grid(True)
plt.legend()
plt.title("Représentation graphique de g(x)")
plt.show()

#Algorithme N°2
import numpy as np
import matplotlib.pyplot as plt

def g(x):
    return -x**2 - 2*x - 1

# Domaine de valeurs
x = np.linspace(-5, 5, 400)
y = g(x)

# Tracé
plt.plot(x, y, label="g(x) = -x² - 2x - 1")
plt.axhline(0)
plt.axvline(0)
plt.grid(True)
plt.legend()
plt.title("Représentation graphique de g(x)")
plt.show()

#Algorithme N°3
import numpy as np
import matplotlib.pyplot as plt

def h(x):
    return -x**2 - 2*x + 13

# Domaine de valeurs
x = np.linspace(-5, 5, 400)
y = h(x)
```

```
# Tracé
plt.plot(x, y, label="h(x) = -x² - 2x + 13")
plt.axhline(0)
plt.axvline(0)
plt.grid(True)
plt.legend()
plt.title("Représentation graphique de h(x)")
plt.show()
```

```
#Algorithme N°4
import numpy as np
import matplotlib.pyplot as plt
```

```
def k(x):
    return x**2 + 2*x - 13
```

```
# Domaine de valeurs
x = np.linspace(-5, 5, 400)
y = K(x)
```

```
# Tracé
plt.plot(x, y, label="k(x) = x² + 2x - 13")
plt.axhline(0)
plt.axvline(0)
plt.grid(True)
plt.legend()
plt.title("Représentation graphique de k(x)")
plt.show()
```