

## Метод Ділення Навпіл

```
import numpy as np

import math

import matplotlib.pyplot as plt

def f(x):
    return x ** 4 - 4 * x - 1

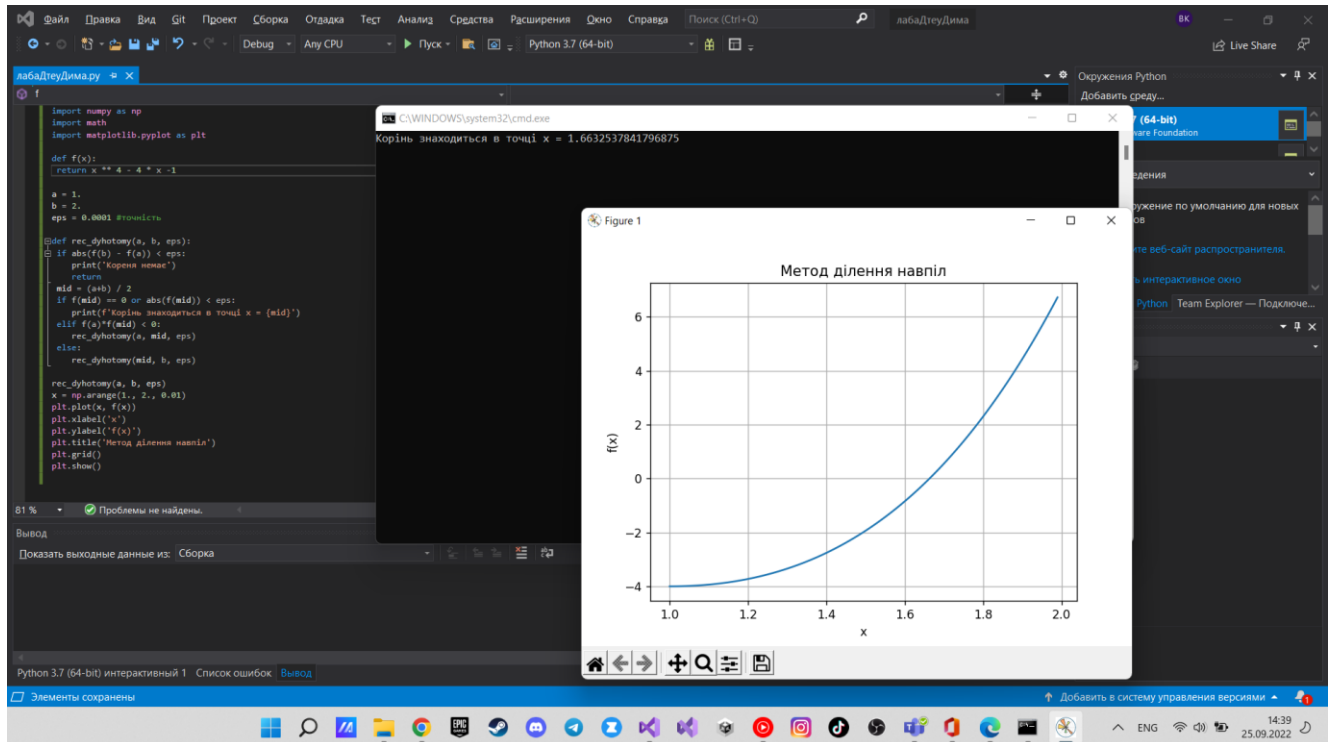
a = 1.
b = 2.
eps = 0.0001 #точність

def rec_dyhotomy(a, b, eps):
    if abs(f(b) - f(a)) < eps:
        print('Кореня немає')
        return
    mid = (a+b) / 2
    if f(mid) == 0 or abs(f(mid)) < eps:
        print(f'Корінь знаходиться в точці x = {mid}')
    elif f(a)*f(mid) < 0:
        rec_dyhotomy(a, mid, eps)
    else:
        rec_dyhotomy(mid, b, eps)

rec_dyhotomy(a, b, eps)

x = np.arange(1., 2., 0.01)
plt.plot(x, f(x))
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Метод ділення навпіл')
plt.grid()
```

plt.show()



## Методом Хорди

from scipy.misc import derivative

def f(x):

return x \*\* 4 - 4 \* x - 1

def hord (a, b, eps):

if abs(f(b) - f(a)) < eps:

print('Нет корней')

return

if (f(a) \* derivative(f, a, n=2)):

x0 = a

xi = b

else:

x0 = b

xi = a

```
xi_1 = xi - (xi - x0) * f(xi) / (f(xi) - f(x0))
```

```
while (abs(f(xi_1) - f(xi)) > eps):
```

```
    xi = xi_1
```

```
    xi_1 = xi - (xi - x0) * f(xi) / (f(xi) - f(x0))
```

```
else:
```

```
    print(f'Корень находится в точке x = ', xi_1)
```

```
hord(1.5, -0.5, 0.0001)
```

