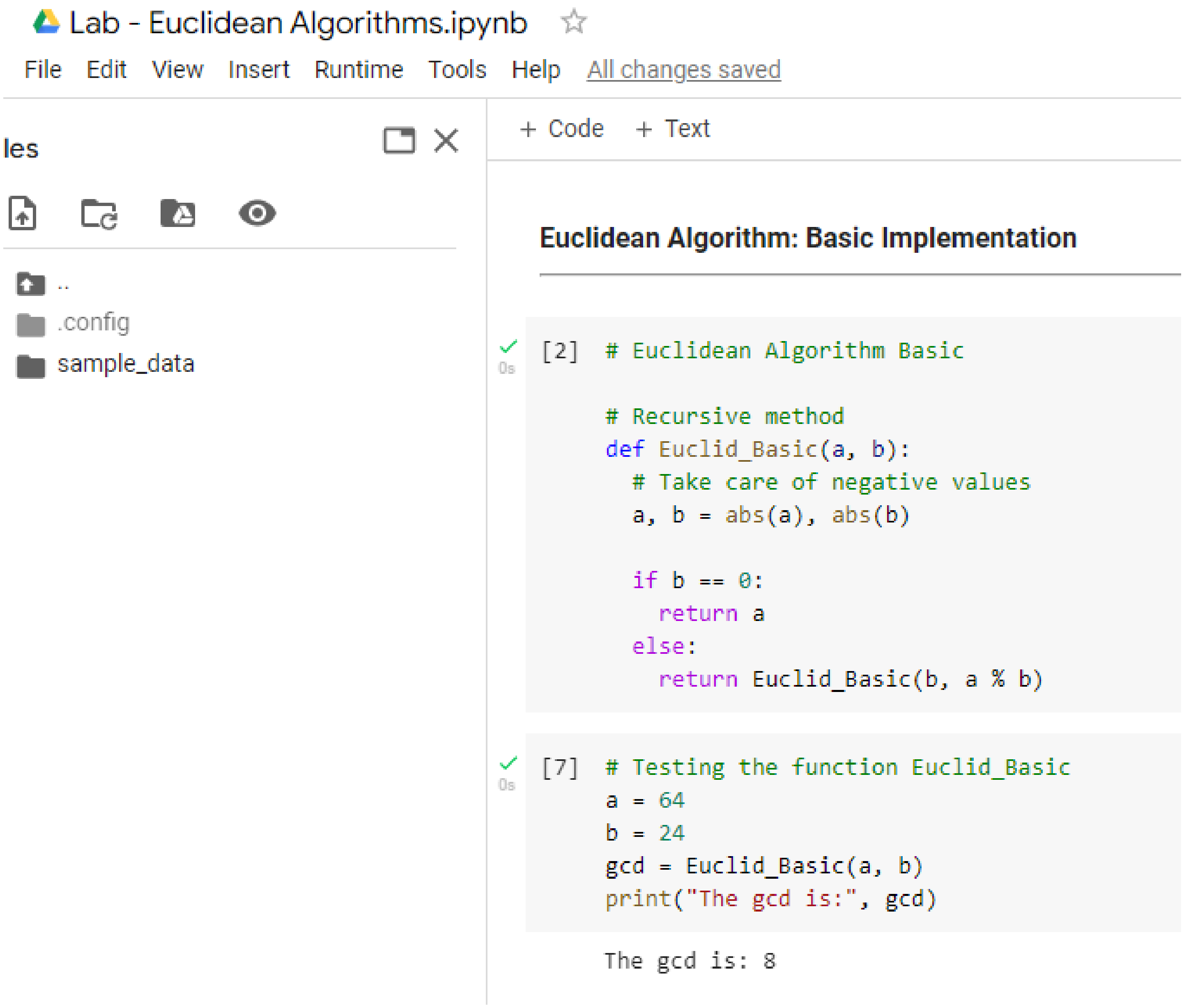
## Lab – Euclidean Algorithms (Basic and Extended)

From the book, we are presented with a basic Euclidean algorithm (recursive method), which I used to implement the code below on Google Colab and tested it on some values and printed their outcomes.



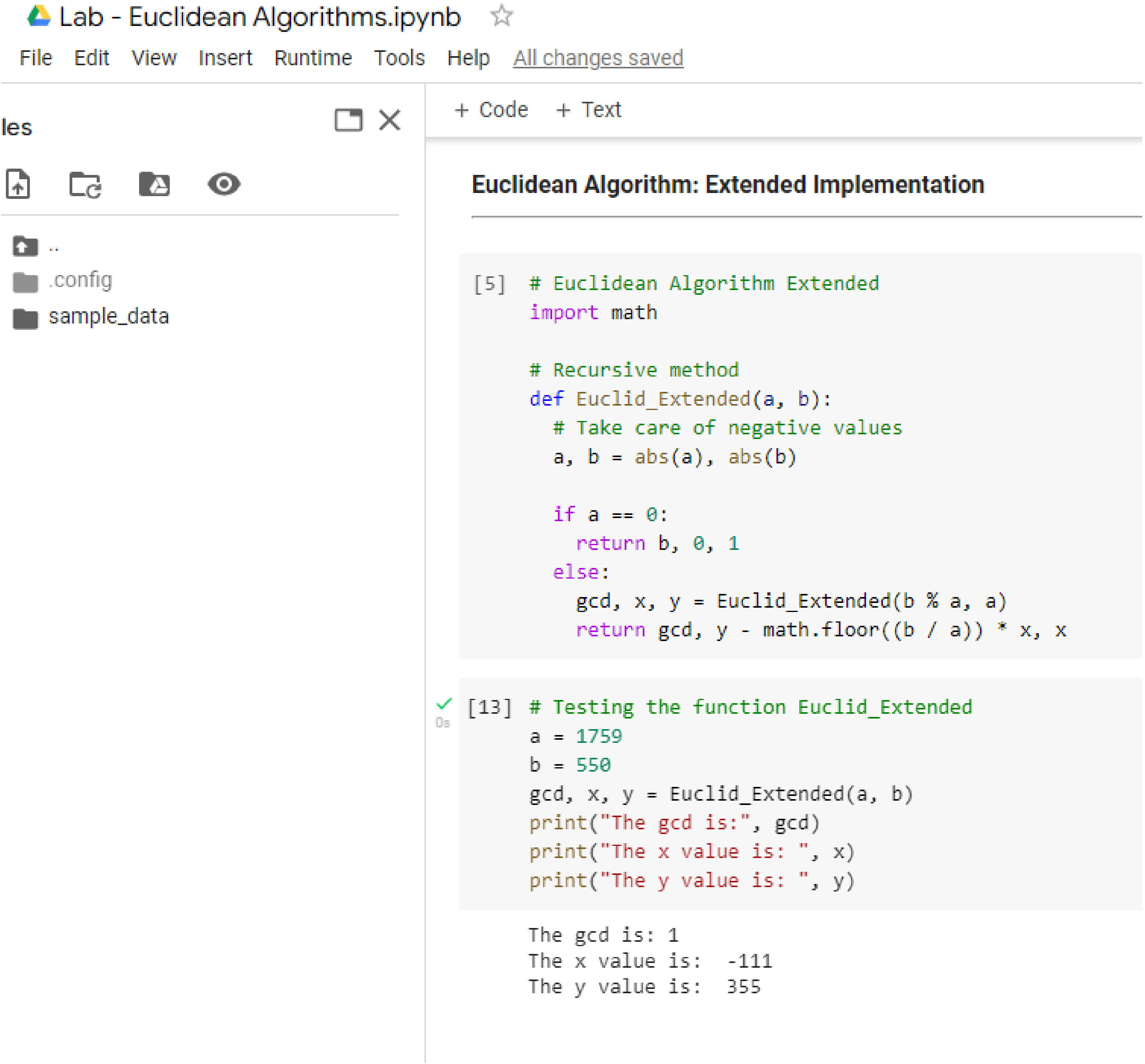
### Euclidean Algorithm: Basic Implementation

```python
# Euclidean Algorithm Basic

# Recursive method
def Euclid_Basic(a, b):
  # Take care of negative values
  a, b = abs(a), abs(b)

  if b == 0:
    return a
  else:
    return Euclid_Basic(b, a % b)
```

```python
# Testing the function Euclid_Basic
a = 64
b = 24
gcd = Euclid_Basic(a, b)
print("The gcd is:", gcd)
```

```
The gcd is: 8
```

For the extended Euclidean algorithm, the book explained it very well, but I still faced trouble in implementing it correctly; therefore, the geeksforgeeks link helped me resolve that problem. I printed the outcomes on the Google Colab shown below.



Euclidean Algorithm: Extended Implementation

```
# Euclidean Algorithm Extended
import math

# Recursive method
def Euclid_Extended(a, b):
  # Take care of negative values
  a, b = abs(a), abs(b)

  if a == 0:
    return b, 0, 1
  else:
    gcd, x, y = Euclid_Extended(b % a, a)
    return gcd, y - math.floor((b / a)) * x, x
```

```
# Testing the function Euclid_Extended
a = 1759
b = 550
gcd, x, y = Euclid_Extended(a, b)
print("The gcd is:", gcd)
print("The x value is: ", x)
print("The y value is: ", y)
```

```
The gcd is: 1
The x value is:  -111
The y value is:  355
```

**References:**

Pearson Education. (2013). *Cryptography and network security: Principles and practice*.

https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/