

**Lab 3: Fake Currency –  
Division, Modulus, and Conditionals**

Due: 11:59pm Saturday, October 4th

Total points: 4

In this lab, you'll experiment with the floor division (`//`) and modulo (`%`) operators. You'll also practice writing conditional statements with `if` and `if-else` patterns.

In Thonny, create a file called `Lab3Exercises.py`.

**Exercise 1:** Type this code in and try several inputs to confirm that it works as you expect it to.

```
num = int(input("Enter your favorite integer: "))  
  
if num % 2 == 0:  
    print(num, "is even")  
else:  
    print(num, "is odd")
```

**Exercise 2:** Replace the `==` in the above code with `=` and run it again. Answer the following questions (for your notes):

- What kind of error does this give you? Why do you think Python gives that error?
- What is the difference between `==` and `=`?

**Exercise 3:** The date June 4, 2024 is a *magic date* because, when written in this format: 6/4/24, the month times the day equals the two-digit year:  $6*4=24$ . Write a program that asks the user to enter a month, a day, and a two-digit year and displays whether or not that date is magic. It should work like this:

```
>>> %Run Lab3Exercises.py  
Enter a number for a month (1-12): 6  
Enter a number for the day (1-31): 4  
Enter a two-digit year: 24  
That is a magic date!
```

```
>>> %Run Lab3Exercises.py  
Enter a number for a month (1-12): 9  
Enter a number for the day (1-31): 3  
Enter a two-digit year: 24  
That is not a magic date.
```

**Exercise 4:** You can include conditional statements inside of other conditional statements like in the example below. This is called a **nested conditional** statement because one of the conditional statements is *nested* inside the other.

Type in this code and run it with different inputs. Would could you do if you wanted it to be able to tell you whether or not a number is even *and* divisible by 3?

```
# enter an integer
num = int(input("please enter an integer: "))

#this code checks if a number is divisible by both 2 and 3
if num % 2 == 0:
    print(num,"is even")
else:
    if num % 3 == 0:
        print(num,"is odd and divisble by 3")
    else:
        print(num,"is odd and not divisible by 3")
```

Create a new file called Lab3.py. make sure your name is in comments at the top of your code.

#### Challenge Exercise:

*Credit: ChatGPT was used to help develop the backstory for this lab. Click [here](#) for reference.*

Consider a fictitious country called Urnestria. Urnestria is a small but culturally rich nation, officially recognized as an independent country in 2006.

The official currency of Urnestria is called the **Urni**. See image below for an example Urni bill. *Note that the person depicted is an AI(ChatGPT) representation of the country's founder.*



Figure 1: a counterfeit Uni currency

Unfortunately, the country of Urnestria was plagued with a large money counterfeit dilemma. In order to combat this, they developed a simple scheme to help authenticate bills based on the year and serial number printed on the bill.

Your objective is to write a program that will determine if a Urni bill is LEGITIMATE or COUNTERFEIT based on the bill's year and printed serial number.

If the bill's year is less than 2006, it is obviously COUNTERFEIT.  
If the bill's year is greater than 2025, it is also COUNTERFEIT.

If the calendar year is **even** (e.g. 2024) then check the serial number  
If the serial number is divisible by 3, then it is LEGITIMATE  
If the serial number is NOT divisible by 3, then it is COUNTERFEIT

However, if the calendar year is **odd** (e.g. 2023) then check serial number  
If the serial number is divisible by 5, then it is LEGITIMATE  
If the serial number is NOT divisible by 5, then it is COUNTERFEIT

Write your program that will prompt the user for the year and serial number. The program should always prompt the user for both numbers. The output should be consistent with the algorithm depicted above. Here are some sample outputs:

```
>>> %Run Lab3.py
Enter a year: 2024
Enter the serial number: 123
LEGITIMATE
```

```
>>> %Run Lab3.py
Enter a year: 2021
Enter the serial number: 1012341934
COUNTERFEIT
```

```
>>> %Run Lab3.py
Enter a year: 2025
Enter the serial number: 125
LEGITIMATE
```

```
>>> %Run Lab3.py
Enter a year: 2026
Enter the serial number: 130
COUNTERFEIT
```

```
>>> %Run Lab3.py
Enter a year: 2023
Enter the serial number: 104595
LEGITIMATE
```

```
>>> %Run Lab3.py
Enter a year: 2022
Enter the serial number: 963813
LEGITIMATE
```

```
>>> %Run Lab3.py
Enter a year: 2009
Enter the serial number: 47
COUNTERFEIT
```

```
>>> %Run Lab3.py
Enter a year: 2010
Enter the serial number: 40
COUNTERFEIT
```

When you are finished, submit your solution to the **Lab 3: Fake Currency** assignment on codePost. This problem will be auto-graded, and it will be checking that your output matches the sample above *exactly*. So, if you think your solution is correct, but codePost doesn't give you the points, check the messages it is giving you - there may be a difference in spelling, punctuation, etc.