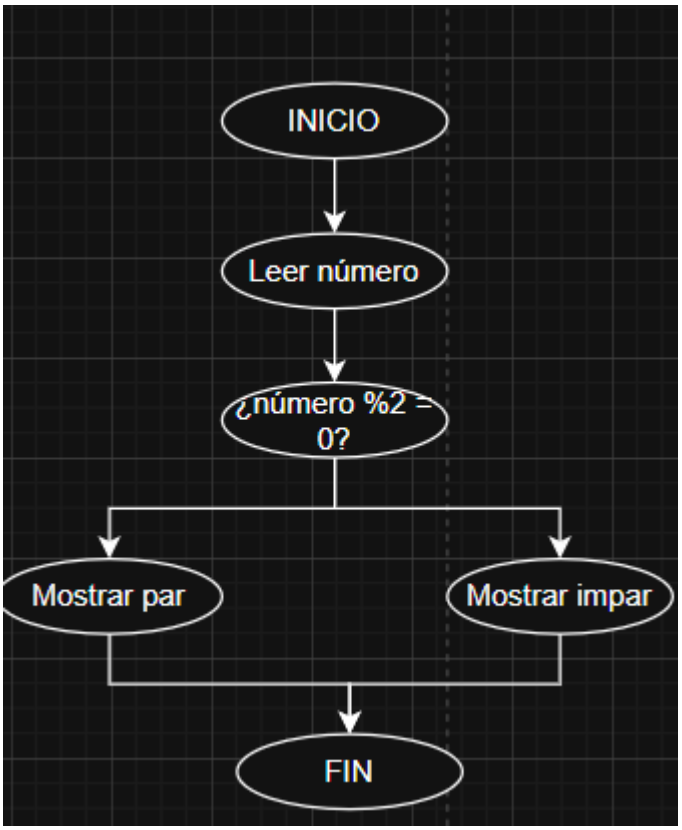
Exercise 34: Even or Odd?

Write a program that reads an integer from the user. Then your program should display a message indicating whether the integer is even or odd.

```
num = int(input("Introduce un número: "))
if num % 2 == 0:
    print("El número es par")
else:
    print("El número es impar")
```

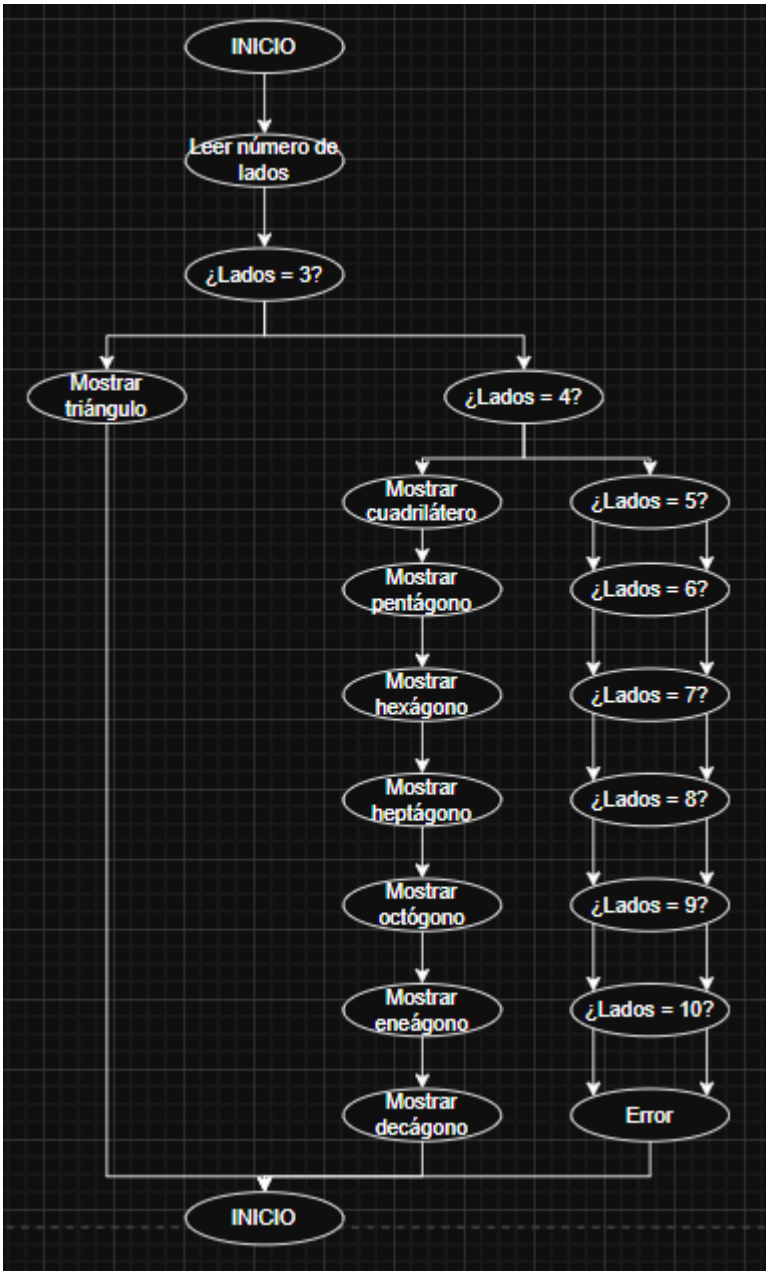


Exercise 37: Name that Shape

Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

```
lados = int(input("Introduce el número de lados: "))
if lados == 3:
    print("Triángulo")
elif lados == 4:
    print("Cuadrilátero")
elif lados == 5:
```

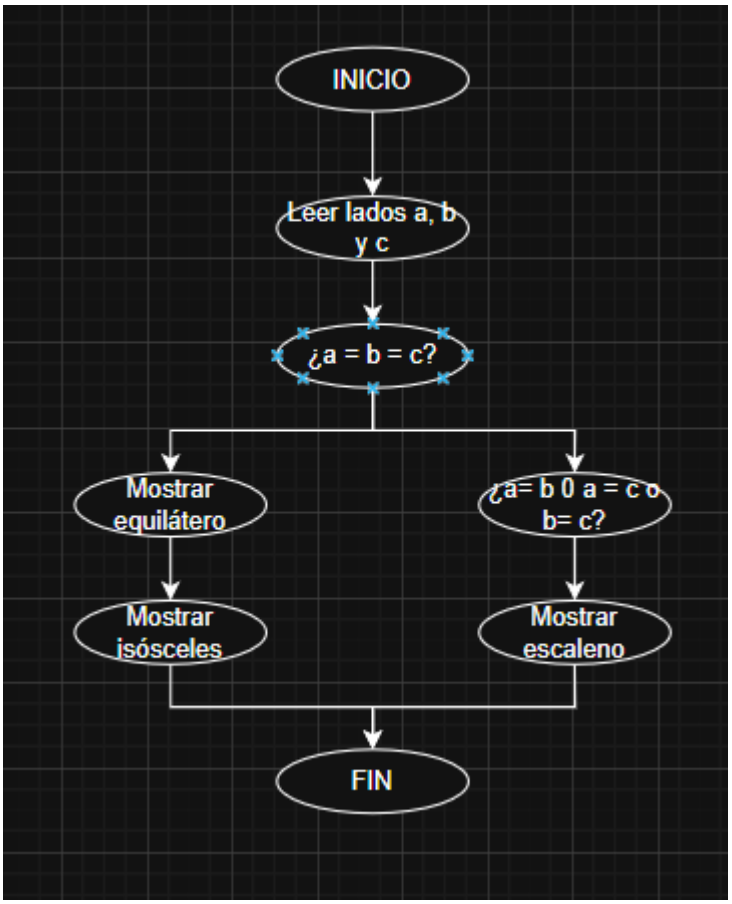
```
    print("Pentágono")
elif lados == 6:
    print("Hexágono")
elif lados == 7:
    print("Heptágono")
elif lados == 8:
    print("Octágono")
elif lados == 9:
    print("Eneágono")
elif lados == 10:
    print("Decágono")
else:
    print("Error: solo se pueden leer valores de 3 a 10 lados")
```



Exercise 40: Name that Triangle

A triangle can be classified based on the lengths of its sides as equilateral, isosceles or scalene. All 3 sides of an equilateral triangle have the same length. An isosceles triangle has two sides that are the same length, and a third side that is a different length. If all of the sides have different lengths then the triangle is scalene. Write a program that reads the lengths of 3 sides of a triangle from the user. Display a message indicating the type of the triangle.

```
a = float(input("Lado 1: "))
b = float(input("Lado 2: "))
c = float(input("Lado 3: "))
if a == b == c:
    print("Equilátero")
elif a == b or a == c or b == c:
    print("Iisósceles")
else:
    print("Escaleno")
```



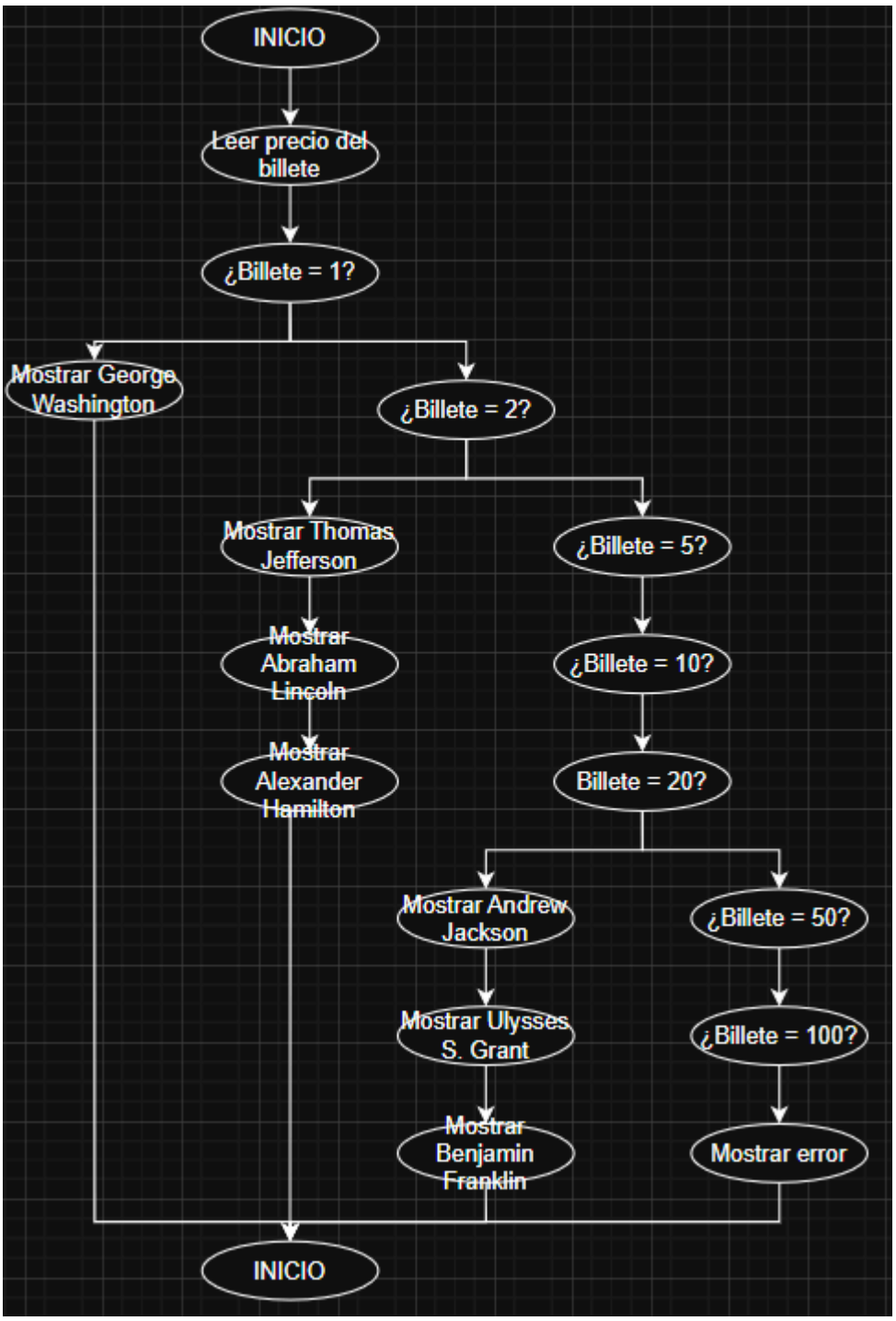
Exercise 43: Faces on Money

It is common for images of a country's previous leaders, or other individuals of his torical significance, to appear on its money.Theindividualsthatappearonbanknotes in the United States are listed in Table 2.1. Write a program that begins by reading the denomination of a banknote from the user. Then your program should display the name of the individual that appears on the banknote of the entered amount. An appropriate error message should be displayed if no such note exists.
While two dollar banknotes are rarely seen in circulation in the United States, they are legal tender that can be spent just like any other denomination. The United States has also issued banknotes in denominations of $500, $1,000, $5,000, and $10,000 for public use. However, high denomination banknotes have not been printed since 1945 and were officially discontinued in 1969. As a result, we will not consider them in this exercise.

Table 2.1 Individuals that appear on Banknotes

Individual	Amount
George Washington	$1
Thomas Jefferson	$2
Abraham Lincoln	$5
Alexander Hamilton	$10
Andrew Jackson	$20
Ulysses S. Grant	$50
Benjamin Franklin	$100

```
billete = int(input("Introduce el precio del billete: "))
if billete == 1:
    print("George Washington")
elif billete == 2:
    print("Thomas Jefferson")
elif billete == 5:
    print("Abraham Lincoln")
elif billete == 10:
    print("Alexander Hamilton")
elif billete == 20:
    print("Andrew Jackson")
elif billete == 50:
    print("Ulysses S. Grant")
elif billete == 100:
    print("Benjamin Franklin")
else:
    print("Error: no hay ningún billete con ese precio")
```



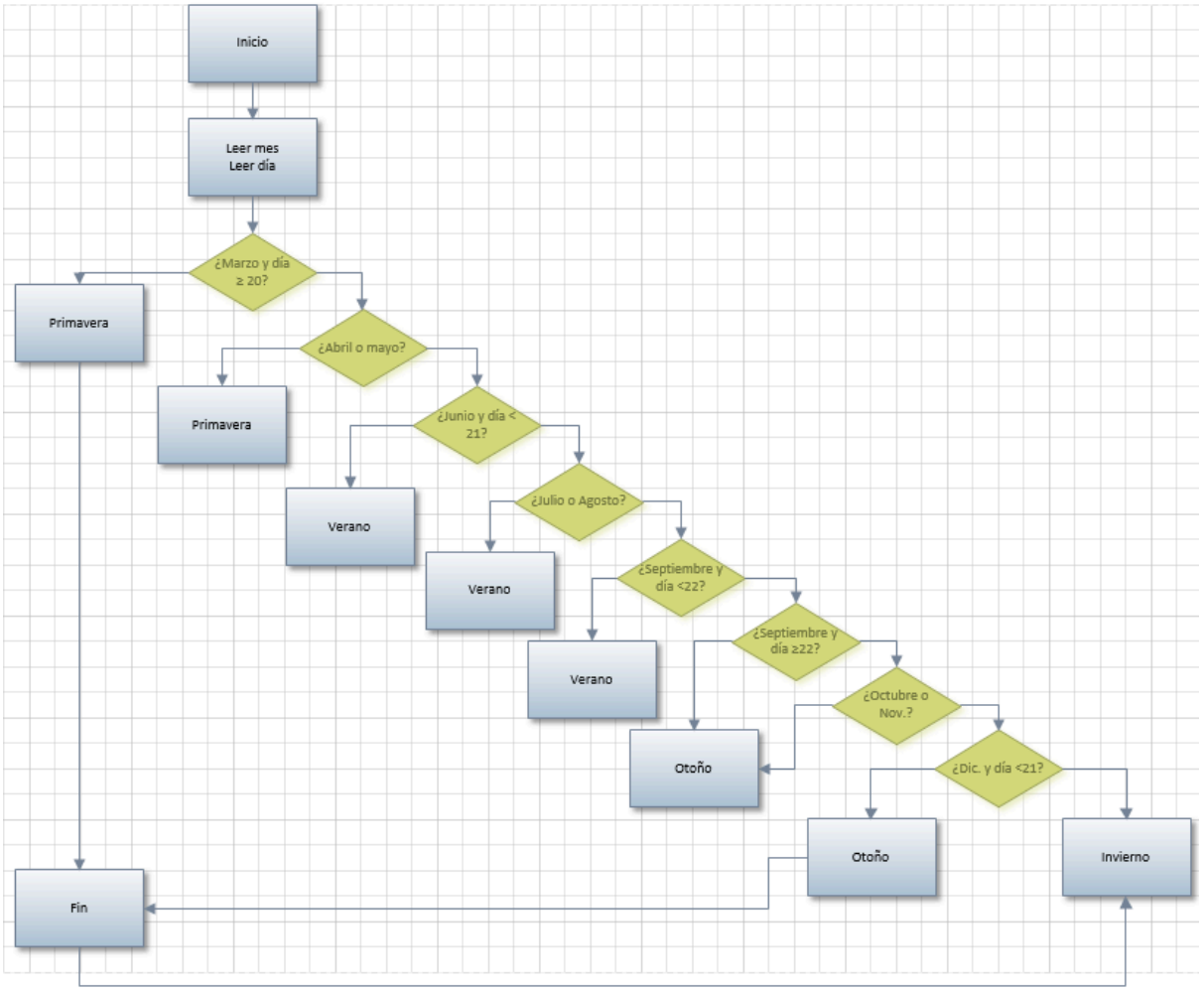
Exercise 46: Season from Month and Day

The year is divided into four seasons: spring, summer, fall and winter. While the exact dates that the seasons change vary a little bit from year to year because of the way that the calendar is constructed, we will use the following dates for this exercise:

Season	First day
Spring	March 20
Summer	June 21
Fall	September 22
Winter	December 21

Create a program that reads a month and day from the user. The user will enter the name of the month as a string, followed by the day within the month as an integer. Then your program should display the season associated with the date that was entered.

```
mes = input("Pon un mes: ").lower()
dia = int(input("Pon un día: "))
if (mes == "marzo" and dia >= 20):
    estacion = "Primavera"
elif (mes == "abril" and "mayo"):
    estacion = "Primavera"
elif (mes == "junio" and dia < 21):
    estacion = "Primavera"
elif (mes == "junio" and dia >= 21):
    estacion = "Verano"
elif (mes == "julio" and "agosto"):
    estacion = "Verano"
elif (mes == "septiembre" and dia < 22):
    estacion = "Verano"
elif (mes == "septiembre" and dia >= 22):
    estacion = "Otoño"
elif (mes == "octubre" and "noviembre"):
    estacion = "Otoño"
elif (mes == "diciembre" and dia < 21):
    estacion = "Otoño"
else:
    estacion = "Invierno"
print("La estación es:", estacion)
```


Inicio
Leer mes
Leer día
¿Marzo y día ≥ 20?
Primavera
¿Abril o mayo?
Primavera
¿Junio y día < 21?
Verano
¿Julio o Agosto?
Verano
¿Septiembre y día <22?
Verano
¿Septiembre y día ≥22?
Otoño
¿Octubre o Nov.?
¿Dic. y día <21?
Otoño
Invierno
Fin

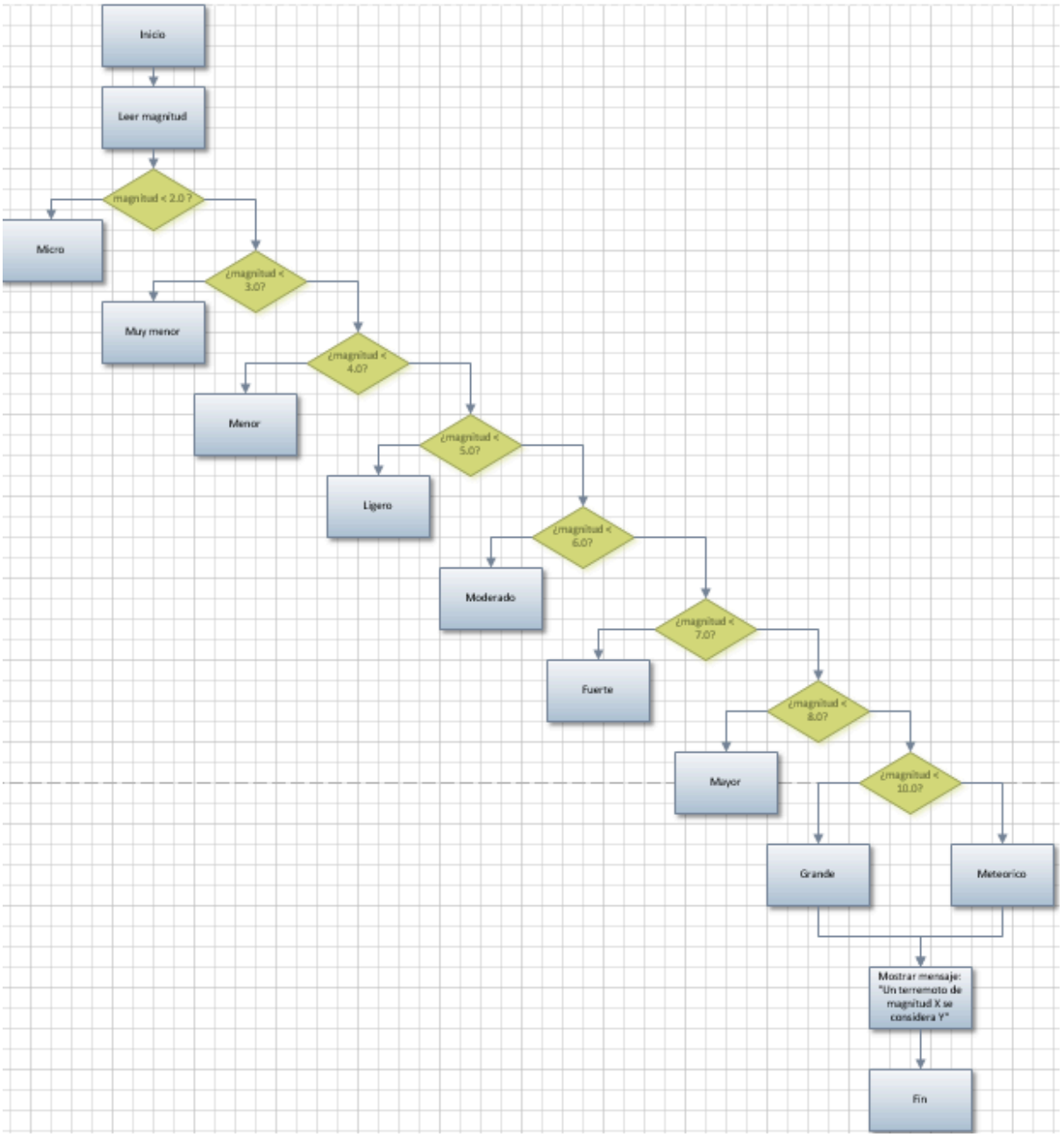
Exercise 49: Richter Scale

The following table contains earthquake magnitude ranges on the Richter scale and their descriptors:

Magnitude	Descriptor
Less than 2.0	Micro
2.0 to less than 3.0	Very minor
3.0 to less than 4.0	Minor
4.0 to less than 5.0	Light
5.0 to less than 6.0	Moderate
6.0 to less than 7.0	Strong
7.0 to less than 8.0	Major
8.0 to less than 10.0	Great
10.0 or more	Meteoric

Write aprogramthatreadsamagnitudefromtheuseranddisplaystheappropriate descriptor as part of a meaningful message. For example, if the user enters 5.5 then your program should indicate that a magnitude 5.5 earthquake is considered to be a moderate earthquake.

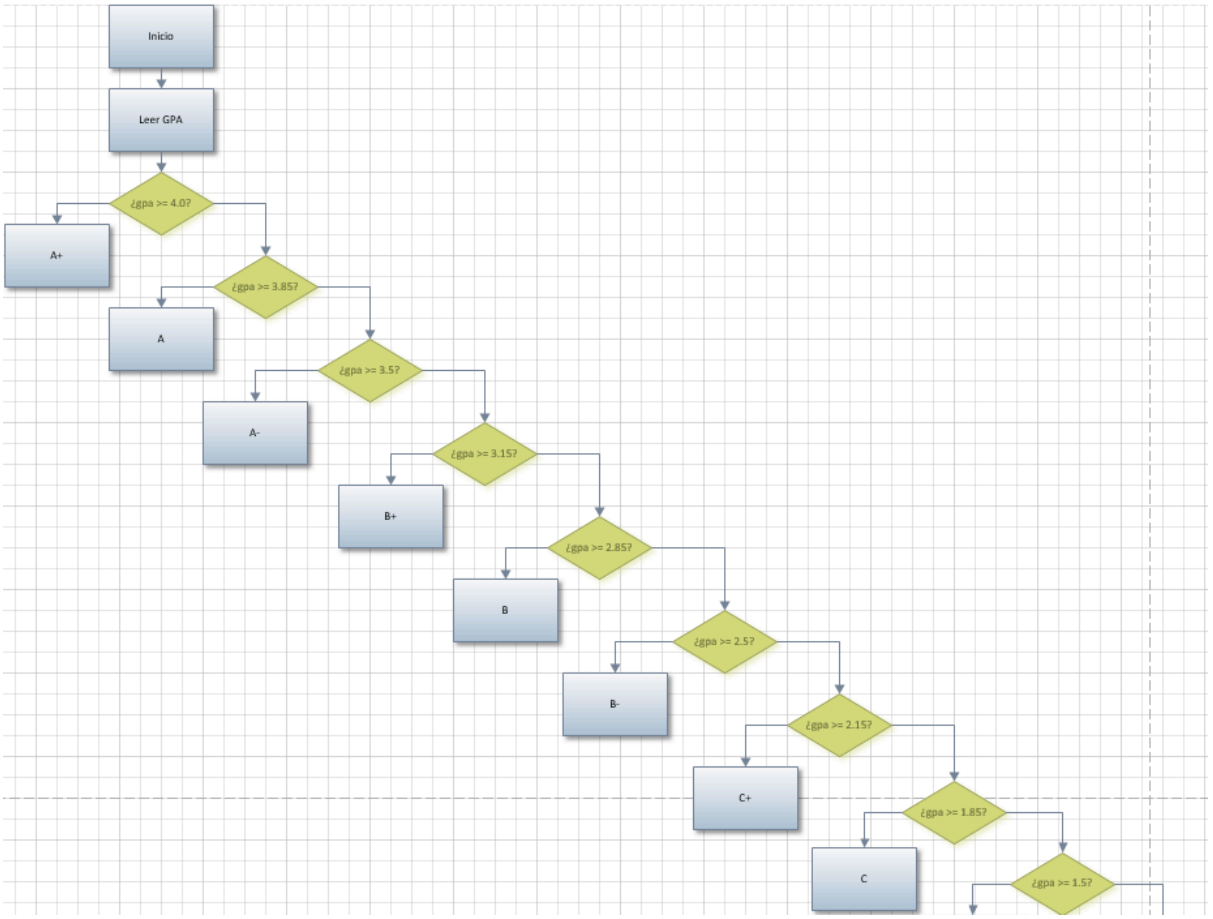
```
magnitud = float(input("Pon la magnitud del terremoto: "))
if magnitud < 2.0:
    descriptor = "Micro"
elif magnitud < 3.0:
    descriptor = "Muy menor"
elif magnitud < 4.0:
    descriptor = "Menor"
elif magnitud < 5.0:
    descriptor = "Ligero"
elif magnitud < 6.0:
    descriptor = "Moderado"
elif magnitud < 7.0:
    descriptor = "Fuerte"
elif magnitud < 8.0:
    descriptor = "Mayor"
elif magnitud < 10.0:
    descriptor = "Grande"
else:
    descriptor = "meteórico"
print(f"Un terremoto de magnitud {magnitud} se considera {descriptor}.")
```

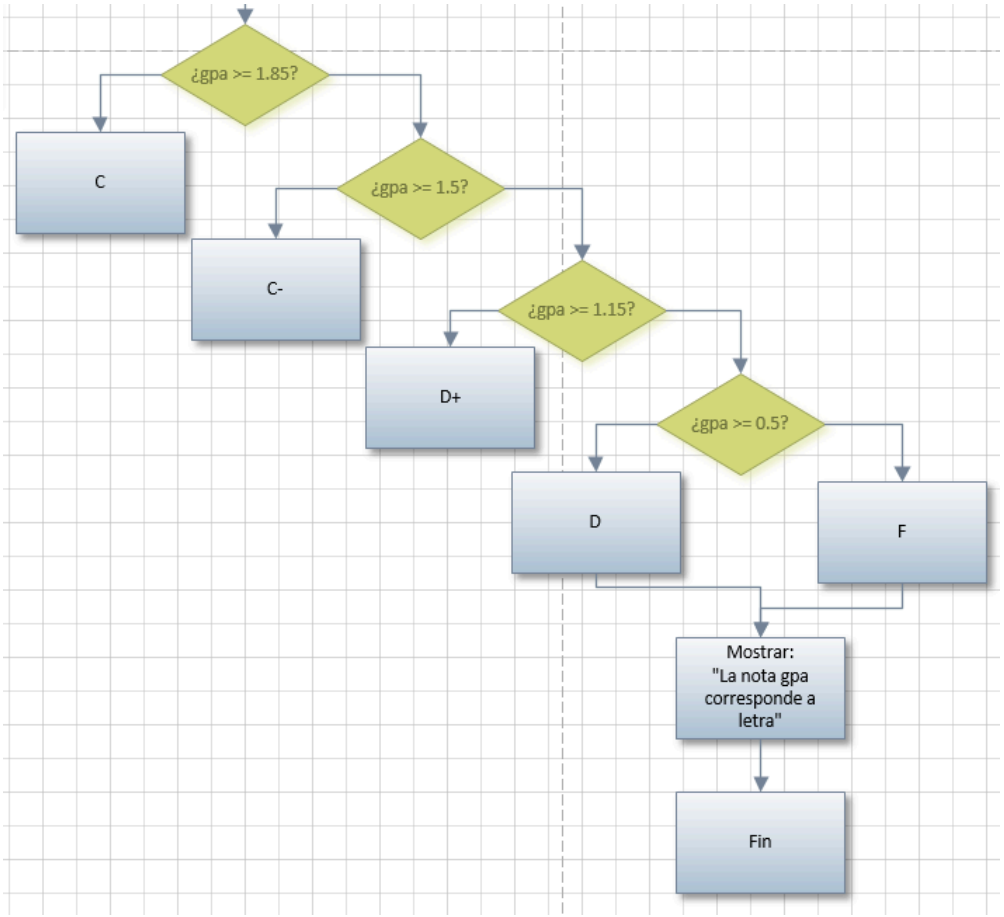

Inicio
Leer magnitud
magnitud < 2.0 ?
Micro
¿magnitud < 3.0?
Muy menor
¿magnitud < 4.0?
Menor
¿magnitud < 5.0?
Ligero
¿magnitud < 6.0?
Moderado
¿magnitud < 7.0?
Fuerte
¿magnitud < 8.0?
Mayor
¿magnitud < 10.0?
Grande
Meteorico
Mostrar mensaje: "Un terremoto de magnitud X se considera Y"
Fin

Exercise 52: Grade Points to Letter Grade

In the previous exercise you created a program that converts a letter grade into the equivalent number of grade points. In this exercise you will create a program that reverses the process and converts from a grade point value entered by the user to a letter grade. Ensure that your program handles grade point values that fall between letter grades. These should be rounded to the closest letter grade. Your program should report A+ for a 4.0 (or greater) grade point average.

```
gpa = float(input("Pon la nota del examen: "))
if gpa >= 4.0:
    letra = "A+"
elif gpa >= 3.85:
    letra = "A"
elif gpa >= 3.5:
    letra = "A-"
elif gpa >= 3.15:
    letra = "B+"
elif gpa >= 2.85:
    letra = "B"
elif gpa >= 2.5:
    letra = "B-"
elif gpa >= 2.15:
    letra = "C+"
elif gpa >= 1.85:
    letra = "C"
elif gpa >= 1.5:
    letra = "C-"
elif gpa >= 1.15:
    letra = "D+"
elif gpa >= 0.5:
    letra = "D"
else:
    letra = "F"
print(f"La nota {gpa} corresponde a {letra}.")
```


Inicio
Leer GPA
¿gpa >= 4.0?
A+
¿gpa >= 3.85?
A
¿gpa >= 3.5?
A-
¿gpa >= 3.15?
B+
¿gpa >= 2.85?
B
¿gpa >= 2.5?
B-
¿gpa >= 2.15?
C+
¿gpa >= 1.85?
C
¿gpa >= 1.5?


¿gpa >= 1.85?
C
¿gpa >= 1.5?
C-
¿gpa >= 1.15?
D+
¿gpa >= 0.5?
D
F
Mostrar:
"La nota gpa
corresponde a
letra"
Fin

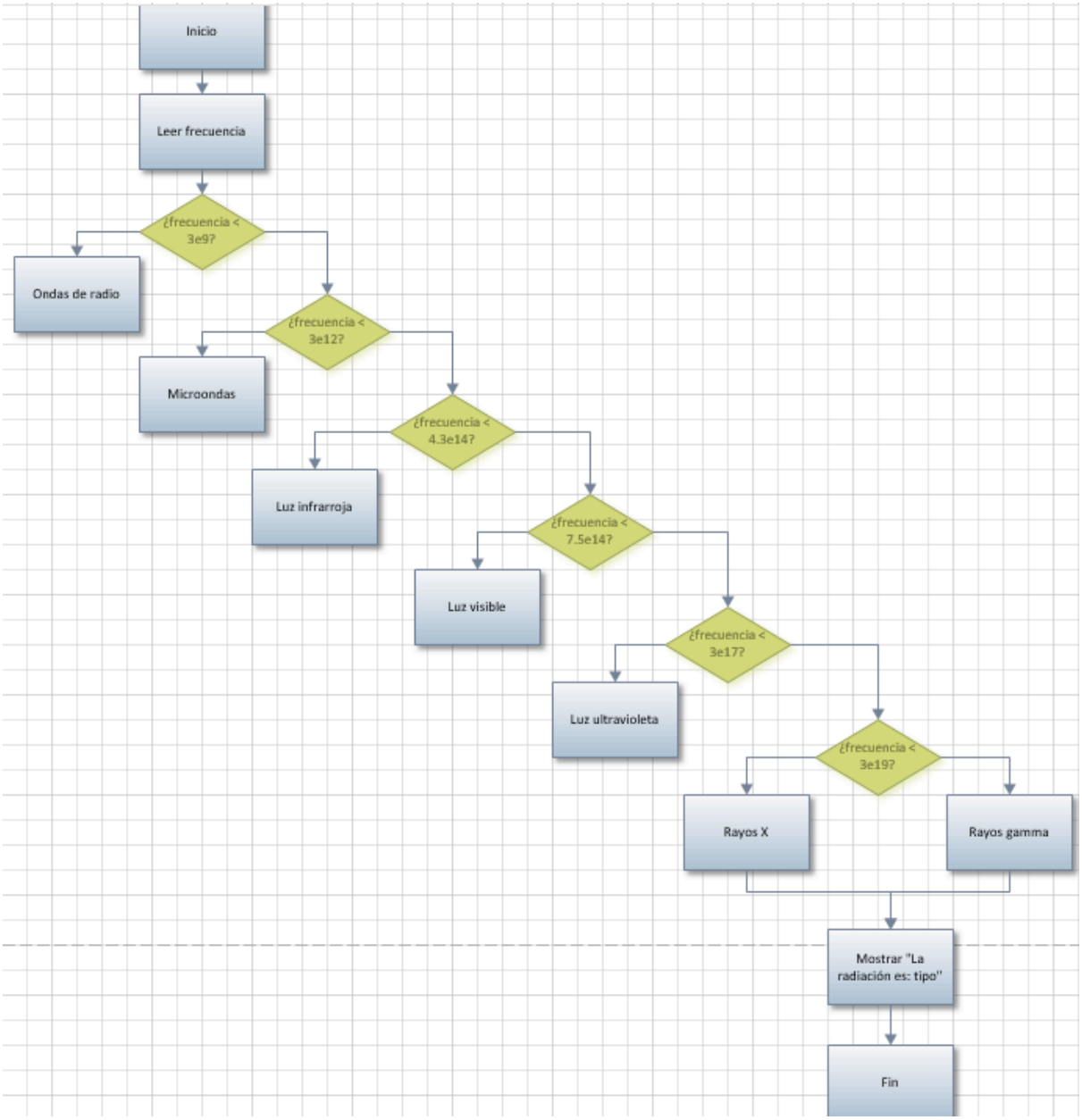
Exercise 55: Frequency to Name

Electromagnetic radiation can be classified into one of 7 categories according to its frequency, as shown in the table below:

Name	Frequency range (Hz)
Radio waves	Less than 3×10^9
Microwaves	3×10^9 to less than 3×10^{12}
Infrared light	3×10^{12} to less than 4.3×10^{14}
Visible light	4.3×10^{14} to less than 7.5×10^{14}
Ultraviolet light	7.5×10^{14} to less than 3×10^{17}
X-rays	3×10^{17} to less than 3×10^{19}
Gamma rays	3×10^{19} or more

Write a program that reads the frequency of the radiation from the user and displays the appropriate name.

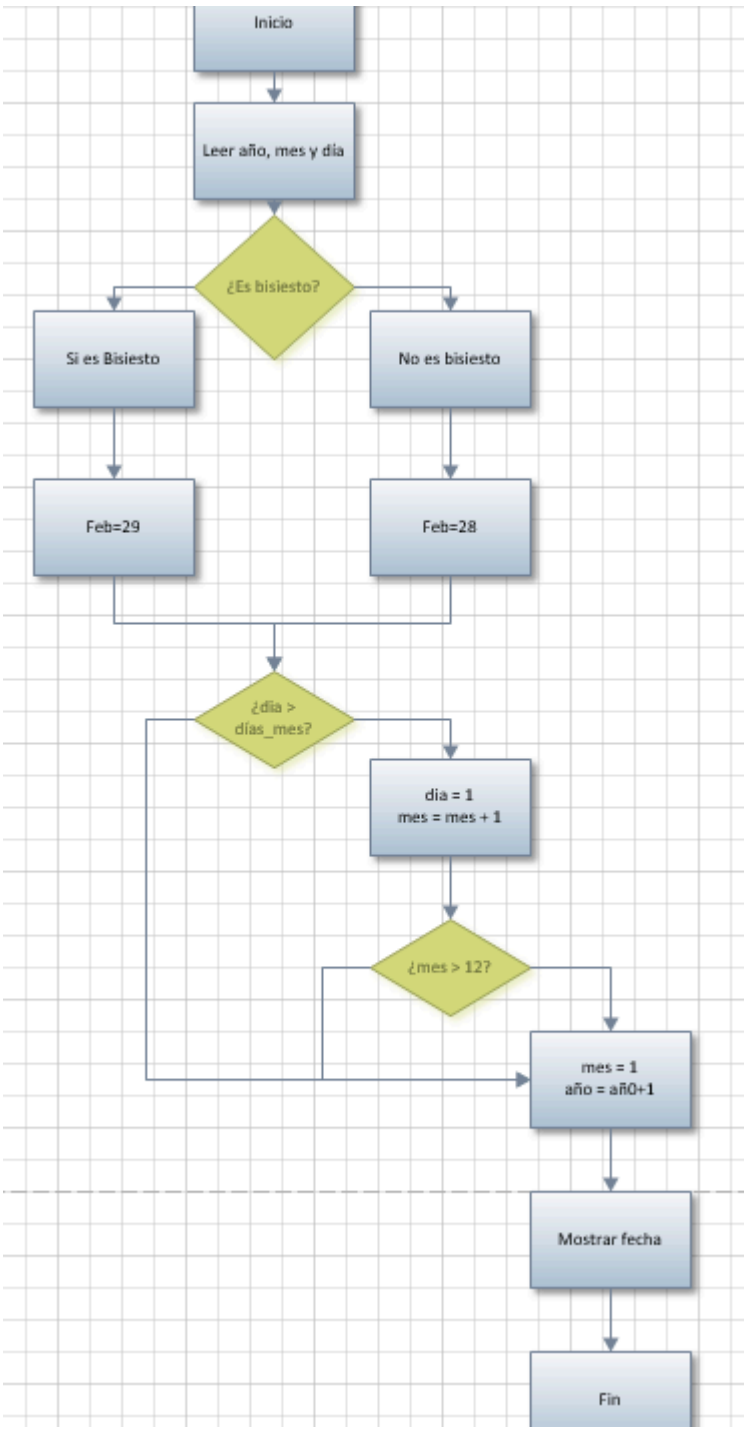
```
frecuencia = float(input("Pon la frecuencia: "))
if frecuencia < 3e9:
    tipo = "Ondas de radio"
elif frecuencia < 3e12:
    tipo = "Microondas"
elif frecuencia < 4.3e14:
    tipo = "Luz infrarroja"
elif frecuencia < 7.5e14:
    tipo = "Luz visible"
elif frecuencia < 3e17:
    tipo = "Luz ultravioleta"
elif frecuencia < 3e19:
    tipo = "Rayos X"
else:
    tipo = "Rayos gamma"
print("La radiación es:", tipo)
```


Inicio
Leer frecuencia
¿frecuencia < 3e9?
Ondas de radio
¿frecuencia < 3e12?
Microondas
¿frecuencia < 4.3e14?
Luz infrarroja
¿frecuencia < 7.5e14?
Luz visible
¿frecuencia < 3e17?
Luz ultravioleta
¿frecuencia < 3e19?
Rayos X
Rayos gamma
Mostrar "La radiación es: tipo"
Fin

Exercise 58: Next Day

Write a program that reads a date from the user and computes its immediate successor. For example, if the user enters values that represent 2013-11-18 then your program should display a message indicating that the day immediately after 2013-11-18 is 2013-11-19. If the user enters values that represent 2013-11-30 then the program shouldindicatethatthenextdayis2013-12-01.Iftheuserentersvaluesthatrepresent 2013-12-31 then the program should indicate that the next day is 2014-01-01. The date will be entered in numeric form with three separate input statements; one for the year, one for the month, and one for the day. Ensure that your program works correctly for leap years.

```
año = int(input("Pon el año: "))
mes = int(input("Pon el mes: "))
dia = int(input("Pon el día: "))
if (año % 400 == 0) or (año % 4 == 0 and año % 100 != 0):
    bisiesto = True
else:
    bisiesto = False
dias_mes = {
    1: 31,
    2: 28 if bisiesto else 29,
    3: 31,
    4: 30,
    5: 31,
    6: 30,
    7: 31,
    8: 31,
    9: 30,
    10: 31,
    11: 30,
    12: 31
}
dia += 1
if dia > dias_mes[mes]:
    dia = 1
    mes += 1
    if mes > 12:
        mes = 1
        año += 1
print(f"El día es {año}-{mes:02d}-{dia:02d}")
```


Inicio
Leer año, mes y dia
¿Es bisiesto?
Si es Bisiesto
No es bisiesto
Feb=29
Feb=28
¿dia > días_mes?
dia = 1
mes = mes + 1
¿mes > 12?
mes = 1
año = añ0+1
Mostrar fecha
Fin