

The programming constructs that you used to solve the exercises in the previous chapter will continue to be useful as you tackle these problems. In addition, the exercises in this chapter will require you to use decision making constructs so that your programs can handle a variety of different situations that might arise. You should expect to use some or all of these Python features when completing these problems:

- Make a decision with an if statement
- Select one of two alternatives with an if-else statement
- Select from one of several alternatives by using an if-elif or if-elif-else statement
- Construct a complex condition for an if statement that includes the Boolean operators `and`, `or` and `not`
- Nest an if statement within the body of another if statement

Exercise 34: Even or Odd?

(Solved—13 Lines)

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.

Exercise 35: Dog Years

(22 Lines)

It is commonly said that one human year is equivalent to 7 dog years. However this simple conversion fails to recognize that dogs reach adulthood in approximately two years. As a result, some people believe that it is better to count each of the first two human years as 10.5 dog years, and then count each additional human year as 4 dog years.

Write a program that implements the conversion from human years to dog years described in the previous paragraph. Ensure that your program works correctly for conversions of less than two human years and for conversions of two or more human years. Your program should display an appropriate error message if the user enters a negative number.

Exercise 36: Vowel or Consonant

(Solved—16 Lines)

In this exercise you will create a program that reads a letter of the alphabet from the user. If the user enters a, e, i, o or u then your program should display a message indicating that the entered letter is a vowel. If the user enters y then your program should display a message indicating that sometimes y is a vowel, and sometimes y is a consonant. Otherwise your program should display a message indicating that the letter is a consonant.

Exercise 37: Name that Shape

(Solved—31 Lines)

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.

Exercise 38: Month Name to Number of Days

(Solved—18 Lines)

The length of a month varies from 28 to 31 days. In this exercise you will create a program that reads the name of a month from the user as a string. Then your program should display the number of days in that month. Display “28 or 29 days” for February so that leap years are addressed.

Exercise 39: Sound Levels

(30 Lines)

The following table lists the sound level in decibels for several common noises.

Noise	Decibel level (dB)
Jackhammer	130
Gas lawnmower	106
Alarm clock	70
Quiet room	40

Write a program that reads a sound level in decibels from the user. If the user enters a decibel level that matches one of the noises in the table then your program should display a message containing only that noise. If the user enters a number of decibels between the noises listed then your program should display a message indicating which noises the level is between. Ensure that your program also generates reasonable output for a value smaller than the quietest noise in the table, and for a value larger than the loudest noise in the table.

Exercise 40: Name that Triangle

(Solved—20 Lines)

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.

Exercise 41: Note To Frequency

(Solved—39 Lines)

The following table lists an octave of music notes, beginning with middle C, along with their frequencies.

Note	Frequency (Hz)
C4	261.63
D4	293.66
E4	329.63
F4	349.23
G4	392.00
A4	440.00
B4	493.88

Begin by writing a program that reads the name of a note from the user and displays the note's frequency. Your program should support all of the notes listed previously.

Once you have your program working correctly for the notes listed previously you should add support for all of the notes from C0 to C8. While this could be done by adding many additional cases to your if statement, such a solution is cumbersome, inelegant and unacceptable for the purposes of this exercise. Instead, you should exploit the relationship between notes in adjacent octaves. In particular, the frequency of any note in octave n is half the frequency of the corresponding note in octave $n + 1$. By using this relationship, you should be able to add support for the additional notes without adding additional cases to your if statement.

Hint: To complete this exercise you will need to extract individual characters from the two-character note name so that you can work with the letter and the octave number separately. Once you have separated the parts, compute the frequency of the note in the fourth octave using the data in the table above. Then divide the frequency by 2^{4-x} , where x is the octave number entered by the user. This will halve or double the frequency the correct number of times.

Exercise 42: Frequency To Note

(Solved—40 Lines)

In the previous question you converted from note name to frequency. In this question you will write a program that reverses that process. Begin by reading a frequency from the user. If the frequency is within one Hertz of a value listed in the table in the previous question then report the name of the note. Otherwise report that the frequency does not correspond to a known note. In this exercise you only need to consider the notes listed in the table. There is no need to consider notes from other octaves.

Exercise 43: Faces on Money

(31 Lines)

It is common for images of a country's previous leaders, or other individuals of historical significance, to appear on its money. The individuals that appear on banknotes 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

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

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.

Exercise 44: Date to Holiday Name

(18 Lines)

Canada has three national holidays which fall on the same dates each year.

Holiday	Date
New year's day	January 1
Canada day	July 1
Christmas day	December 25

Write a program that reads a month and day from the user. If the month and day match one of the holidays listed previously then your program should display the holiday's name. Otherwise your program should indicate that the entered month and day do not correspond to a fixed-date holiday.

Canada has two additional national holidays, Good Friday and Labour Day, whose dates vary from year to year. There are also numerous provincial and territorial holidays, some of which have fixed dates, and some of which have variable dates. We will not consider any of these additional holidays in this exercise.