Fundamentals of Python: First Programs

Chapter 4: Number Systems

Objectives

After completing this lesson, you will be able to

- Convert a string representation of a number from one base to another base

Problem

Convert 637_8 (Octal = base 8) to a decimal number (base 10)

Today’s Lesson “The How”

- Reading a “Technical Text”
- Determine the central ideas of the text, summarizing the complex concepts, processes, and/or information presented in the text by paraphrasing them in simpler but still accurate terms.
- Determine the meaning of symbols, key terms, and Python commands.

Today's Lesson "The What"

- Number Systems: Converting hexadecimal, octal, binary, and decimal numbers
- Read Section 4.3 (Pages 129 – 135)

Today’s Lesson “The How Part 2”

- – Start with private thinking time.
- – We will use “Listen & Compare” Structured Discussion with your partner.
- – Groups will share
  - What you learned including:
  - The positional system
  - Converting Binary to Decimal
  - Converting Decimal to Binary
  - Converting Hexadecimal & Octal to Binary using the shortcuts

Problem Part 2

Convert 637_8 (Octal = base 8) to a decimal number (base 10)

Exit Ticket

– Socrative.com
– Room number: LCHS607

Number Conversion.xlsx

Strings and Number Systems

415 in binary notation	110011111_2
415 in octal notation	637_8
415 in decimal notation	415_{10}
415 in hexadecimal notation	$19F_{16}$

- The digits used in each system are counted from 0 to *n* - 1, where *n* is the **system's base**
- To represent digits with values larger than 9_{10}, systems such as base 16 use letters
  - Example: A_{16} represents the quantity 10_{10}, whereas 10_{16} represents the quantity 16_{10}

The Positional System for Representing Numbers

- In **positional notation**, a digit has a **positional value**, determined by raising the base to the power specified by the position (*base*^{position})

Positional values	100	10	1
Positions	2	1	0

[FIGURE 4.2] The first three positional values in the base 10 number system

$$415_{10} =$$

$$4 * 10^2 + 1 * 10^1 + 5 * 10^0 =$$

$$4 * 100 + 1 * 10 + 5 * 1 =$$

$$400 + 10 + 5 = 415$$

Converting Binary to Decimal

- Each digit or bit in binary number has positional value that is power of 2
- We occasionally refer to a binary number as a string of bits or a **bit string**
- To determine the integer quantity that a string of bits represents:

$$1100111_2 =$$

$$1 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 =$$

$$1 * 64 + 1 * 32 + 0 * 16 + 0 * 8 + 1 * 4 + 1 * 2 + 1 * 1 =$$

$$64 + 32 + 4 + 2 + 1 = 103$$

Converting Binary to Decimal (continued)

```
"""
File: binarytodecimal.py
Converts a string of bits to a decimal integer.
"""

bstring = input("Enter a string of bits: ")
decimal = 0
exponent = len(bstring) - 1
for digit in bstring:
    decimal = decimal + int(digit) * 2 ** exponent
    exponent = exponent - 1
print("The integer value is", decimal)

> python binarytodecimal.py
Enter a string of bits: 1111
The integer value is 15
> python binarytodecimal.py
Enter a string of bits: 101
The integer value is 5
```

Converting Binary to Decimal (cont.)

```
"""
File: decimaltobinary.py
Converts a decimal integer to a string of bits.
"""

decimal = int(input("Enter a decimal integer: "))
if decimal == 0:
    print (0)
else:
    print("Quotient Remainder Binary")
    bstring = ""
    while decimal > 0:
        remainder = decimal % 2
        decimal = decimal // 2
        bstring = str(remainder) + bstring
        print("%5d%8d%12s" % (decimal, remainder, bstring))
    print("The binary representation is", bstring)

> python decimalToBinary.py
Enter a decimal integer: 34
Quotient Remainder Binary
   17       0           0
    8       1          10
    4       0         010
    2       0        0010
    1       0       00010
    0       1      100010
The binary representation is 100010
```

Conversion Shortcuts

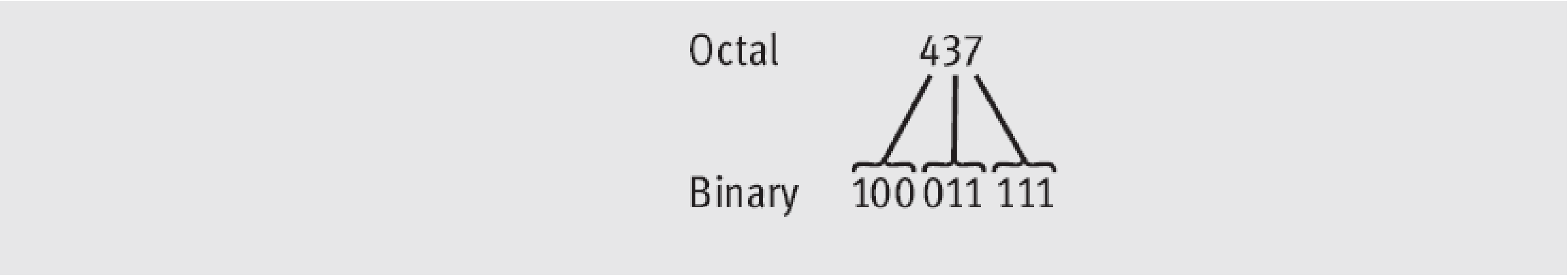
DECIMAL	BINARY
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

[TABLE 4.1] The numbers 0 through 8 in binary

- Thus, a quick way to compute the decimal value of the number 11111_2 is 2^5 - 1, or 31_{10}

Octal and Hexadecimal Numbers

- To convert from octal to binary, start by assuming that each digit in the octal number represents three digits in the corresponding binary number

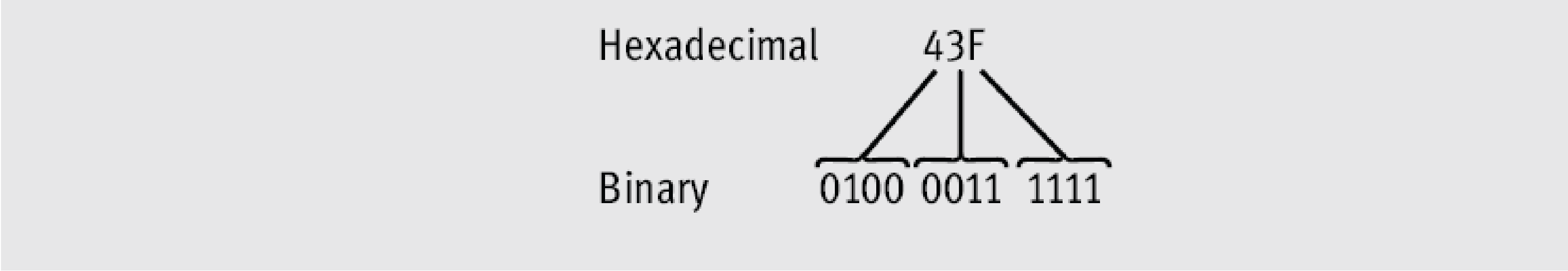


[FIGURE 4.3] The conversion of octal to binary

- To convert binary to octal, you begin at the right and factor the bits into groups of three bits each

Octal and Hexadecimal Numbers (continued)

- To convert from hex to binary, replace each hex digit with the corresponding 4-bit binary number



[FIGURE 4.4] The conversion of hexadecimal to binary

- To convert from binary to hex, factor the bits into groups of 4 and look up the corresponding hex digits