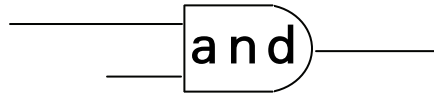


Mathematics for the Digital Age



Programming in Python

>>> Second Edition:
with Python 3

Maria Litvin

Phillips Academy, Andover, Massachusetts

Gary Litvin

Skylight Software, Inc.

Skylight Publishing
Andover, Massachusetts

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: <http://www.skylit.com>
e-mail: sales@skylit.com
support@skylit.com

**Copyright © 2010 by Maria Litvin, Gary Litvin, and
Skylight Publishing**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the authors and Skylight Publishing.

Library of Congress Control Number: 2009913596

ISBN 978-0-9824775-8-8 (soft cover)

ISBN 978-0-9824775-4-0 (hard cover)

The names of commercially available software and products mentioned in this book are used for identification purposes only and may be trademarks or registered trademarks owned by corporations and other commercial entities. Skylight Publishing and the authors have no affiliation with and disclaim any sponsorship or endorsement by any of these products' manufacturers or trademarks' owners.

1 2 3 4 5 6 7 8 9 10 15 14 13 12 11 10

Printed in the United States of America

- ∞ (infinity symbol), 64
- (difference) operator for sets, 252
- != operator, 69
- ' ' ' literal string delimiter, 32, 33
- & (intersection) operator for sets, 252
- & operator, 123, 125
- *= operator, 68
- / operator, 53
- // operator, 53
- ^ (symmetric difference) operator for sets, 253
- ^ operator, 123
- | (union) operator for sets, 252
- | operator, 123
- ~ operator, 123
- + operator for lists, 154
- += operator, 68
- <, >, <=, >= operators, 69
- << and >> operators, 126
- operator, 68
- == operator, 69

- absolute pathname, 161
- add method of set, 253
- adder circuit, 122
- addition rule for probabilities, 233
- adjacency matrix, 280
- adjacent vertices in a graph, 279
- algorithm, 11
- anagram, 137, 166
- analog electronics, 117
- AND circuit, 116
- AND gate, 118
- and operator, 69, 107

- arguments, 2, 51
- arithmetic mean, 54, 59
- arithmetic sequence, 58
 - common difference, 59
- ASCII code, 23
- ASCII file, 160
- Assembly Language, 20
- assignment operator, 38
- augmented assignment operators, 68

- backslash, 33
- base case in recursion, 196
- base-10 numbers, 78
- bin built-in function, 84
- binary arithmetic, 84
- binary files, 160
- binary system, 81
- binomial coefficients, 215
- binomial theorem, 215
- bitwise logical operators, 123
- Blackjack, 231
- Boole, George, 95
- Boolean algebra, 95
- Boolean functions, 109
- bucket (in a hash table), 251
- built-in functions, 39
- bytecode, 25

- carriage return, 160
- Central Processing Unit (CPU), 19
- check digit, 169, 171
- checksum, 171
 - credit cards, 173
 - UPC, 173

- Chinese remainder theorem, 311
- choice function, 237
- Chomp game, 181
- classe, 88
- close method of file, 161
- C_n , 279
- coefficients of a polynomial, 205
- combinations, 131, 140
- comments, 15, 30
- common divisor, 296
- compiler, 25
- complement of a set, 102
- complete graph, 263
- complex numbers, 213
- conditional branching instructions, 20
- conditional jump instructions, 105
- conditional operators, 108
- congruence modulo d , 306
- conjunction, 96
- connected graph, 268, 279
- constants, 40
- converging sequence, 58
- converging series, 64
- corollary, 213, 302
- CPU (Central Processing Unit), 19
 - instructions, 20
 - registers, 21
- Craps game, 233
- cryptology, 313
- cycle (graph), 264

- database of sequences, 58
- De Morgan's laws, 97, 103
- debug program, 20
- debugging, 20
- decimal into binary conversion, 83
- def statement, 32
- default values for arguments, 52
- degree of a vertex in a graph, 270
- del operator for lists, 155
- dictionary, 244, 254

- difference operator for sets, 252
- Diffie-Hellman Key Exchange
 - algorithm, 314
- digital electronics, 117
- Diophantine equation, 297
- directed graphs, 180, 274, 279
- directories, 160
- discard method of set, 253
- discrete logarithm, 316
- disjunction, 96
- diverging sequence, 58
- division operator, 53
- divisor, 301
- divisors of zero, 309
- documentation string, 32, 33
- domain of a function, 3
- dot product of vectors, 248

- edges of a graph, 259, 262
- elements of a set, 2
- elif keyword, 107
- else clause, 106
- empty set, 2, 251
- endswith method of string, 152
- equivalence relation, 267, 269, 306
 - class, 267
- Euclid's algorithm, 296
- Euler circuit, 272, 278
- Euler path, 260, 272
- Euler, Leonhard, 259
- even parity, 169, 170
- exceptions, 40, 73
- exponent (in floating-point numbers), 88

- factor theorem for polynomials, 213
- factorial function, 74
- False and True constants, 107
- Fermat's Last Theorem, 295
- Fermat's Little Theorem, 310, 317

- Fibonacci numbers, 173, 190, 193, 203, 301
- file, 160
 - ASCII, 160
 - binary, 160
 - closing, 161
 - creating, 163
 - extension, 160
 - folder, 160
 - opening, 161
 - output, 163
 - readlines method, 162
 - text, 160
 - write method, 163
- file object type, 162
- find method of string, 149, 150
- finite set, 2
- finite strategy game, 169, 179
- float type, 38
- floating-point numbers, 38
 - IEEE standard, 88
- flowchart, 12
- for loop, 147
- formal parameters, 51
- formatted output, 40
- Four-Color theorem, 287
- Fraction class example, 88
- fully triangulated graph, 288
- function names, 15, 31
- function, 1
 - arguments, 2, 46, 51
 - defined with a formula, 7
 - domain, 3
 - formal parameters, 51
 - natural domain, 7
 - range, 3
 - recursively defined, 189
 - raising an exception, 48
- fundamental theorem of algebra, 213
- fundamental theorem of arithmetic, 301, 304
- garbage collection, 150
- gates, 117, 118
- general term of a sequence, 57
- geometric mean, 59
- geometric sequence, 59
 - common difference, 58
 - sum of, 61
- geometric series, 64
- gigabyte, 19, 20
- gigahertz, 20
- global variables, 44
- Goldbach's conjecture, 305
- golden ratio, 93, 191
- graph, 259
 - adjacency matrix, 280
 - complete, 263
 - connected, 268
 - cycle, 264
 - degree of a vertex, 270
 - directed, 274
 - isomorphism, 266
 - multigraphs, 262
 - optimal path, 275
 - planar, 262
 - properly colored, 284
 - simple, 262
 - subgraph, 269
 - weighted, 275
- greatest common divisor, 296
- Greco-Roman square, 80
- Hamilton circuit, 273
- Hamilton, Sir William Rowan, 273
- harmonic series, 65
- hash table, 244, 251, 254
- hashing, 251
- hex built-in function, 84
- hex digits, 128
- hex into binary conversion, 82
- hex numbers, 124
 - in Python, 83
- hexadecimal system, 21, 82

- IDLE program, 27, 31
- IEEE standard for floating-point numbers, 88
- if-else statement, 105
 - elif keyword, 107
 - else clause, 106
 - with pass, 106
- imaginary number $i = \sqrt{-1}$, 213
- immutable objects, 41, 53, 149, 154, 157
- import statement, 45
- in operator, 108
 - for strings, 148
 - for lists, 154
 - for tuples, 157
- indentation, 14
- indentation, 32
- independent events, 233
- index method of list, 155
- indices, 148
 - negative, 148
- infinite set, 2
- infinity symbol, 64
- input function, 48, 72
- insert method of list, 154
- int built-in function, 84, 123
- int type, 38
- interpreter, 25
- intersection of sets, 101
- intersection operator for sets, 252
- invariant, 169, 175
- irrational numbers, 92
- ISBN-10 and ISBN-13 checkdigit, 174
- isomorphic graphs, 266
- isomorphism, 179
- issubset method of set, 253
- iterations, 69, 190
- iterative statements, 69
- Java, 20
- Kempe's chains, 290
- Keno game, 230
- keyword, 31
- kilobyte, 20
- K_n , 279
- Koch Snowflake, 67
- laws of logic, 97
 - duality, 97
- len built in function, 148, 252
- limit of the sequence, 58
- linear algebra, 243
- linear Diophantine equation, 298
- linear polynomial, 206
- list built-in function, 154
- list comprehensions, 156
- lists, 154
 - reversing, 155
 - sorting, 154
- literal string, 17, 29
- local variables, 43
- logical operators, 69, 107, 108
- long division for polynomials, 212
- loop invariant, 176
- Lucas, Edouard, 200
- Mad Libs program, 167
- magic square, 80, 250
- mantissa, 88
- map, 254
- mapping, 1
- Mastermind game, 159
- math module, 45
- mathematical induction, 62, 135, 200, 288
- matrix, 243, 245
- max built in function, 154, 252
- megabyte, 20
- Mersenne prime, 113
- methods, 48, 149

- min built in function, 154, 252
- mod operator, 305
- Monte Carlo simulations, 238
- motherboard, 19
- multigraphs, 262, 279, 281
- multiplication of polynomials, 210
- multiplication rule, 131
 - for probabilities, 231

- names in Python, 31
 - of variables, 39
- NAND gate, 120
- natural domain, 7
- n-choose-k numbers, 140
- negation, 97
- nested loops, 70, 72
- newline character, 160, 161
- Nim sum, 183
- Nim, 182
- NOR gate, 120
- NOT circuit, 116
- NOT gate, 118
- not in operator, 108
- not logical operator, 69, 107

- octal numbers, 124
 - in Python, 83
- octal system, 81
- odd parity, 170
- OOP (object-oriented programming), 149
- open built-in function, 161
- Open Source license, 27
- operating system, 160
- optimal path (in a weighted graph), 275
- OR circuit, 116
- OR gate, 118
- or logical operator, 69, 107
- output formatting, 40

- parity bit, 169, 170
- parity, 170
- parsing, 28
- partial sum, 64
- Pascal, Blaise, 131, 217
- Pascal's Triangle, 217
- pathname, 161
- perfect number, 75
- permutations, 135
- planar graphs, 262, 284
- polynomials, 205
 - addition, 206
 - coefficients, 205
 - division with a remainder, 211
 - factor theorem, 213
 - linear, 206
 - long division, 212
 - multiplication, 210
 - quadratic, 206
 - remainder theorem, 213
 - roots (zeros), 213
 - subtraction, 208
- pop CPU instruction, 195
- positional number system, 78
- postfix notation, 150
- precedence of operators, 29
- predicate, 99
- prime numbers, 57
- printing to a file, 163
- printed circuit board, 19
- probability of an event, 222
- probability space, 221
- programming language, 24
- prompt, 27
- proof, 8
 - by contradiction, 91, 303
 - irrationality of $\sqrt{2}$, 91
- properly colored graph, 284
- proposition, 95
- pseudocode, 11
- pseudorandom numbers, 236
- push CPU instruction, 195

`py` extension, 71
Pythagorean triple, 92

Q.E.D. (*quod erat demonstrandum*), 9
quadratic polynomial, 206
quadratic spline, 206

RAM (random-access memory), 19
`randint` function, 237
random function, 237
random module, 159
random numbers, 236
random walk, 234
range of a function, 3
rational numbers, 88, 90
`readlines` method of file, 162
recurrence relation, 190
recursion, 194
 base case, 196
recursively defined function, 189
redundancy, 26, 169
references to objects, 41
reflexivity (of a relation), 267
relational operators, 69, 107
relatively prime numbers, 297, 301
relay switch, 115
remainder, 306
remainder theorem for polynomials,
 213
`remove` method of set, 253
`replace` method of string, 151
reserved words, 31
`reverse` method of list, 155
roots of a polynomial, 213
`round` built-in function, 143
RSA algorithm, 317

saving a program in a file, 71
scientific notation, 89

scope of a variable, 43
Selection Sort, 159
sequence, 57
 arithmetic, 58
 as a function, 57
 converging, 58
 general term, 57
 geometric, 59
 limit of, 58
 term, 57

series, 63
 converging, 64
 geometric, 64
 harmonic, 65
 sum of, 64
 telescopic, 66

set, 1, 244
 element, 2
 empty, 2
 finite, 2
 infinite, 2
 subset, 2

Seven Bridges of Königsberg puzzle,
 259

shift operators, 126
short-circuit evaluation, 110
`shuffle` function, 238
sigma notation, 61
simple graph, 262, 279
slice of a list, 154
slice of a sequence, 148
`sort` method of list, 154
sorted list, 158
source code, 27
source file, 71
`sqrt` function from `math`, 45
stack, 22, 195
`stdin` standard input stream, 72
`startswith` method of string, 152
`str` type, 38
strategy stealing, 181

- strings, 147
 - as sequences, 147
 - endswith method, 152
 - find method, 150
 - immutability 149
 - in operator, 148
 - replace method, 151
 - startswith method, 152
 - strip method, 150
 - upper method, 149
 - zfill method, 151
- strip method of string, 150
- subgraph, 269
- subscript, 245
- subset, 2
- substitution cipher, 313
- Sudoku puzzle, 80
- sum built in function, 74, 252
- sum of a series, 64
- sum of vectors, 246
- symmetric difference operator for sets, 253
- symmetry (of a relation), 267
- syntax rules, 14, 26
- system stack, 22

- telescopic sequence, 62
- telescopic series, 66
- term of a sequence, 57
- text file, 160
- topology, 260, 263
- Towers of Hanoi puzzle, 197
- transistor, 19, 117
- transitivity (of a relation), 267
- transposition errors, 171
- True and False constants, 107
- truth table, 96, 97
- tuples, 157
 - immutability, 157
- two's complement representation, 87
- two-dimensional table, 244
- TypeError, 28
- types of values, 28, 37

- Unicode, 148, 160
- union of sets, 102
- union operator for sets, 252
- update method of set, 253
- upper method of string, 149

- ValueError exception, 73
- variable, 37
 - global, 44
 - local, 43
 - scope of, 43
- vectors, 246
 - dot product, 248
 - sum, 246
- Venn diagram, 101
- vertices of a graph, 259, 262
- Vigenere cipher, 319
- von Neumann, John, 117

- weighted graph, 275, 279
- while loop, 69
- Wilson's theorem, 312
- Wolf, Goat, and Cabbage puzzle, 278
- write method of file, 163

- XNOR gate, 120
- XOR gate, 120

- zeros of a polynomial, 213
- zfill method of string, 151

```

# This program demonstrates some of the elements
# of Python's syntax
# Author: H. Dumpty

def someFun(n):
    '''This function takes a positive integer n
       performs some mysterious calculations, and
       returns a positive integer'''
    k = 0
    lst = [] # empty list

    while n > 0:
        if n % 2 != 0:
            lst.append(k) # or: lst += [k]
            k += 1 # same as: k = k + 1
            n //= 2 # integer division with truncation

    lst2 = [2 ** k for k in lst] # list comprehension
    return sum(lst2)

n = -1

while n <= 0:
    s = input('Enter a positive integer: ')
    try:
        n = int(s)
    except ValueError:
        print('Invalid input')

r = someFun(n)
print('n =', n, end=' ')
print('r =', r)

# Three other ways to display the same output:

print('n =', n, 'r =', r)
print('n = {0:d} r = {1:d}'.format(n, r))
print('n = ' + str(n) + ' r = ' + str(r))

```