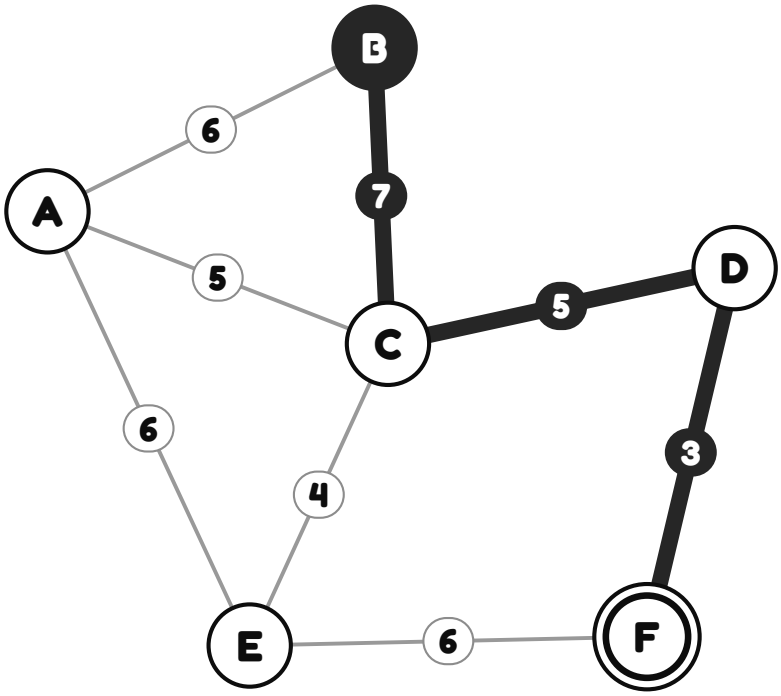
SCREEN-FREE CODING WORKBOOK • AGES 9-12

Algorithm Puzzles for Kids 9-12

99 Screen-Free Coding Puzzles: Flowcharts, Sorting, Binary Search, Shortest Paths, Map Coloring and More, with Lessons and Answers



Harper Lane

Sample pages

Algorithm Puzzles for Kids 9-12

99 Screen-Free Coding Puzzles: Flowcharts, Sorting, Binary Search, Shortest Paths, Map Coloring and More, with Lessons and Answers

Copyright © 2026 Harper Lane. All rights reserved.

No part of this book may be reproduced or shared in any form without written permission from the publisher, except for brief quotations in a review. Parents and teachers may copy single puzzle pages for use with their own children or class.

All 99 puzzles in this book are original. Every answer was worked out by an exact algorithm and checked again by a second, separately written program.

First edition, 2026








This book belongs to

Algorithm Explorer







Contents

For Parents and Teachers 5

My Star Chart 7

 1. Follow the Flowchart 8

 2. Decision Trees 18

 3. Parity Card Magic 28

 4. Binary Search 38

 5. Sorting Networks 48

 6. Sort by Swapping 58

 7. Tower Moves 68

 8. Shortest Path 78

 9. Cheapest Network 88

 10. Color the Map 98

 11. Pack the Bag 108

Your Certificate 118

Answer Key 119

Words to Know 137

Sample pages

For Parents and Teachers

This book has **99 puzzles** in **11 chapters** for children 9-12. Each chapter is one real idea from computer science: following a program, sorting, searching, shortest routes, networks, coloring and packing. No computer, tablet or coding experience is needed, only a pencil.

How the book works

Every chapter opens with **the big idea** in plain words, numbered steps and a **worked example**. Then come the puzzles, from **Warm-Up** (1 star), **Thinker** (2 stars) and **Challenger** (3 stars). The stars on each page show the level. Every page repeats its own short instructions, so a child can open the book anywhere.

Every answer is checked

Each answer was worked out by an exact algorithm (for example Dijkstra's shortest path or Kruskal's cheapest network) and then checked again by a second, separately written program that tries every possibility. When a page asks for "the" route or "the" cables, there is only **one** right answer. When many answers are right (there are many ways to color a map or to order the swaps), the page asks for a **number**, such as the fewest colors or the fewest swaps, and the answer key shows one way to get it.

Tips for helping

- Read the chapter opener together and talk through the worked example before the first puzzle.
- Use a pencil with an eraser. Trying, crossing out and trying again is exactly how algorithms are invented.
- Ask "How do you know it is the shortest?" Explaining a method is the heart of computational thinking.
- Coins, buttons or paper scraps make the sorting and tower puzzles hands-on.
- The answer key starts on page 119, and the star chart on page 7 tracks progress.

Sample pages

Chapter Guide

What each chapter teaches, and where the same idea is used in the real world. The chapters build on each other, but any chapter can be done on its own.



1. Follow the Flowchart
Learns: sequence, selection (if/else), loops, variables
Used for: planning apps and games before coding them



2. Decision Trees
Learns: classification, conditions, binary decisions
Used for: spam filters, medical checklists, game characters



3. Parity Card Magic
Learns: error detection, parity bits, even and odd
Used for: catching errors on disks and in network messages



4. Binary Search
Learns: searching, halving, logarithms (how many times you can halve)
Used for: dictionaries, contact lists, databases



5. Sorting Networks
Learns: comparison, parallel processing, sorting
Used for: fast sorting inside computer chips



6. Sort by Swapping
Learns: sorting algorithms, counting steps (efficiency)
Used for: comparing how fast sorting methods are



7. Tower Moves
Learns: recursion, planning, counting moves
Used for: fractals, folders inside folders, game search



8. Shortest Path
Learns: graphs, shortest path (Dijkstra), why greedy shortcuts fail
Used for: map apps, delivery routes, the internet



9. Cheapest Network
Learns: graphs, spanning trees (Kruskal), greedy algorithms that work
Used for: power lines, pipes, computer networks



10. Color the Map
Learns: graph coloring, constraints, logical deduction
Used for: timetables, radio channels, computer memory



11. Pack the Bag
Learns: optimization, trade-offs, why greedy can fail
Used for: loading trucks, budgets, choosing ads

Sample pages

My Star Chart

Color a star each time you finish a puzzle. Can you collect them all?



Follow the Flowchart 1 2 3 4 5 6 7 8 9



Decision Trees 10 11 12 13 14 15 16 17 18



Parity Card Magic 19 20 21 22 23 24 25 26 27



Binary Search 28 29 30 31 32 33 34 35 36



Sorting Networks 37 38 39 40 41 42 43 44 45



Sort by Swapping 46 47 48 49 50 51 52 53 54



Tower Moves 55 56 57 58 59 60 61 62 63



Shortest Path 64 65 66 67 68 69 70 71 72



Cheapest Network 73 74 75 76 77 78 79 80 81



Color the Map 82 83 84 85 86 87 88 89 90



Pack the Bag 91 92 93 94 95 96 97 98 99

Sample pages

CHAPTER 1

Puzzles 1 to 9

Follow the Flowchart

The Big Idea

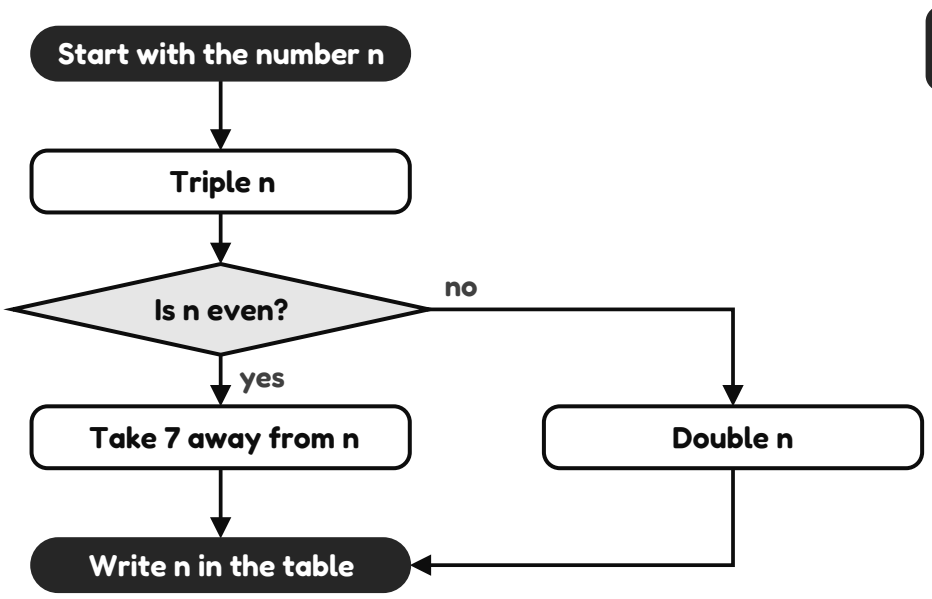
A computer program is a list of exact steps. A **flowchart** draws those steps as boxes joined by arrows. A box tells you to **do** something; a diamond asks a **yes or no** question and sends you one way or the other. Some arrows loop back, so you repeat steps until a question tells you to stop.

How to Solve It

1. Start at the top with the number from the table.
2. In a box, do exactly what it says. The number **n** changes.
3. In a diamond, answer the question and follow **yes** or **no**.
4. When you reach the end, write what comes out.

Worked Example

Here is a small puzzle, already solved.



n	Out
1	6
20	53
14	35

n 1 → **6** · n 20 → **53** · n 14 → **35**

Computers Use It

Every app on a phone is built from these three things: steps, questions and loops. Programmers sketch flowcharts to plan a program before they write the code.

Super Solver Tip

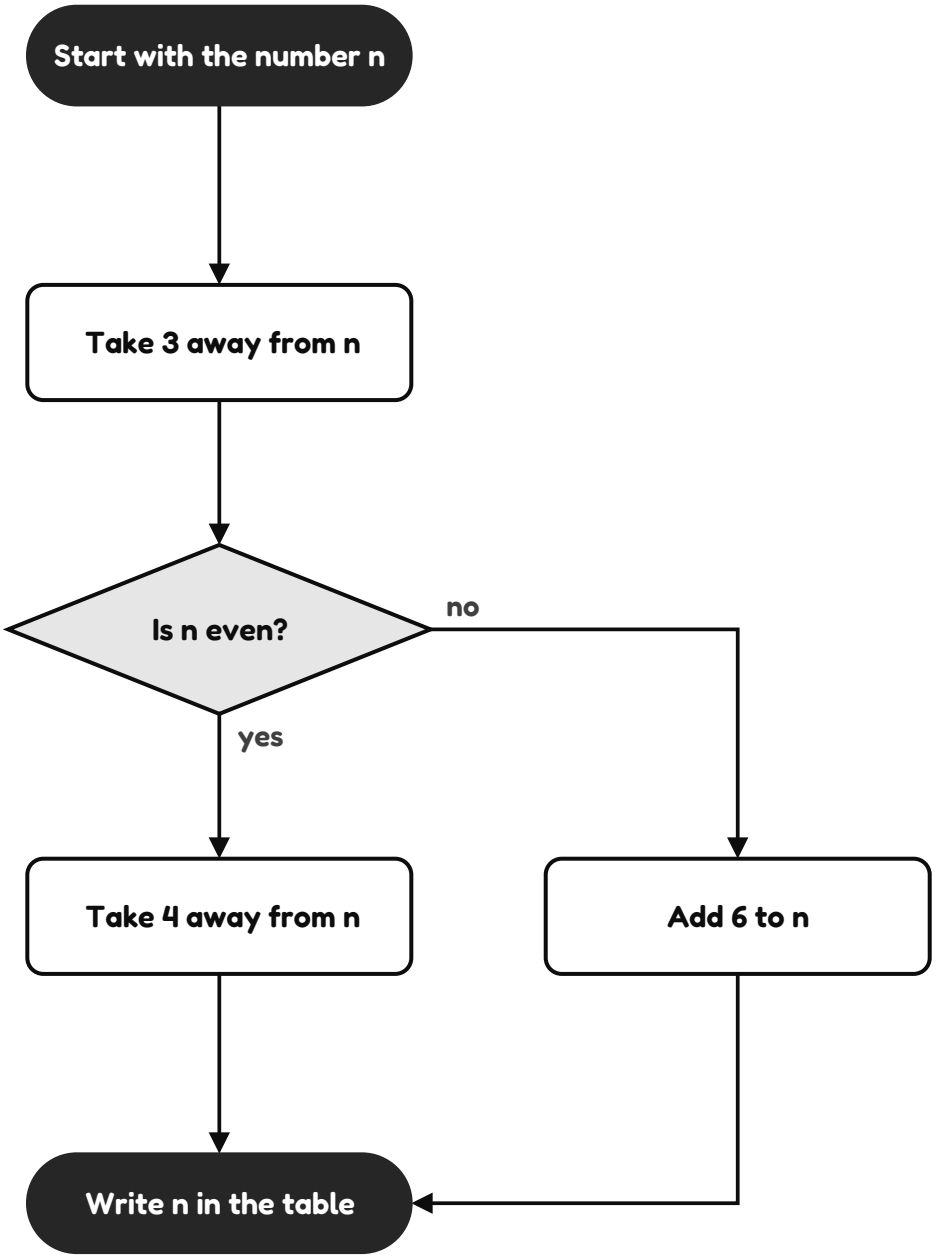
Write the number down every time it changes. It is easy to lose track inside a loop!

Sample pages

Follow the Flowchart

★☆☆ **No. 1**

Follow the arrows, starting at the top, with each starting number **n** from the table. Do what each box says (n changes as you go) and answer each question **yes** or **no**. Write what comes out in the table.



n	Out
9	
14	
19	

Work space

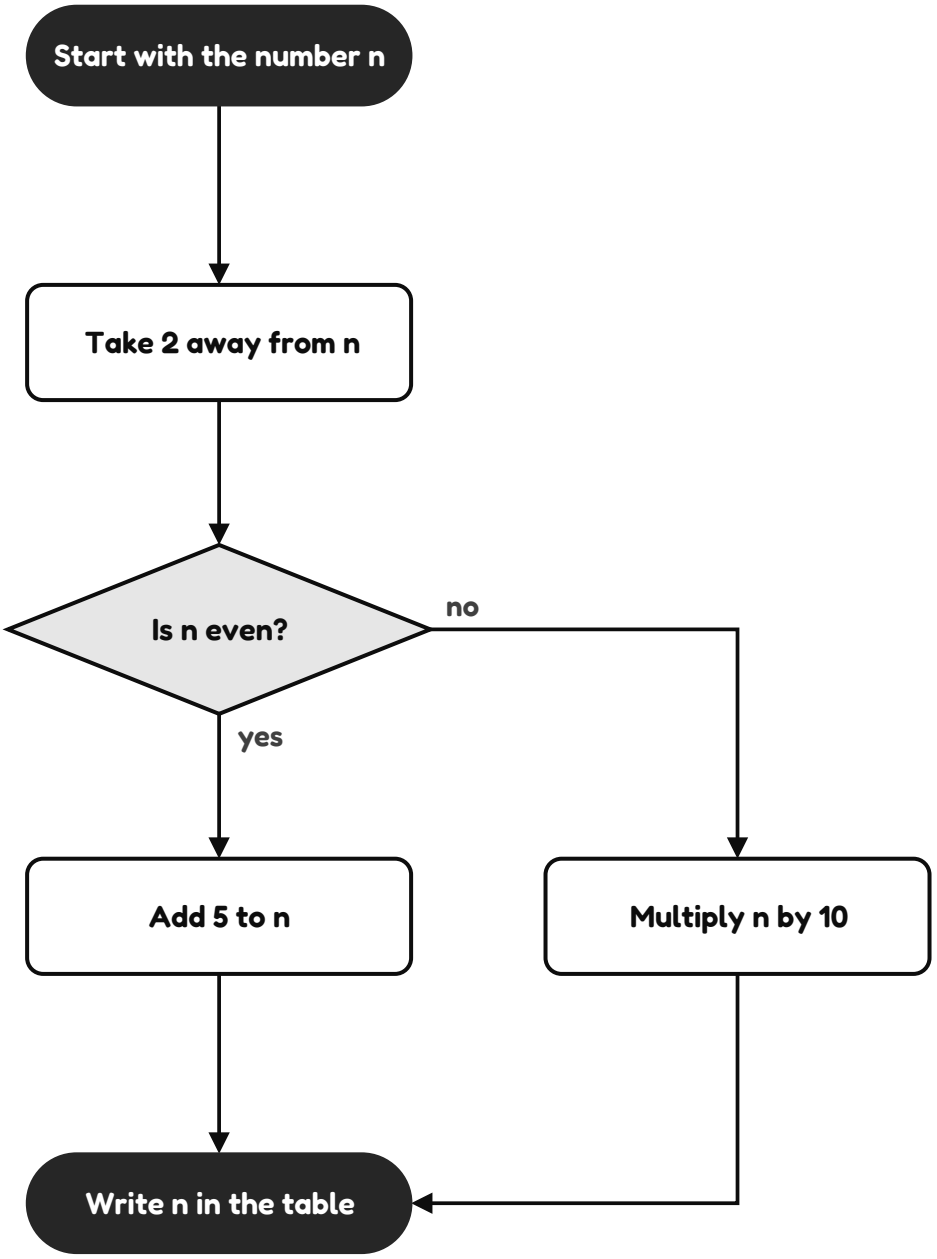
☆ Color the star when you finish!

Sample pages

Follow the Flowchart

★☆☆ **No. 2**

Follow the arrows, starting at the top, with each starting number **n** from the table. Do what each box says (n changes as you go) and answer each question **yes** or **no**. Write what comes out in the table.



n	Out
19	
4	
14	

Work space

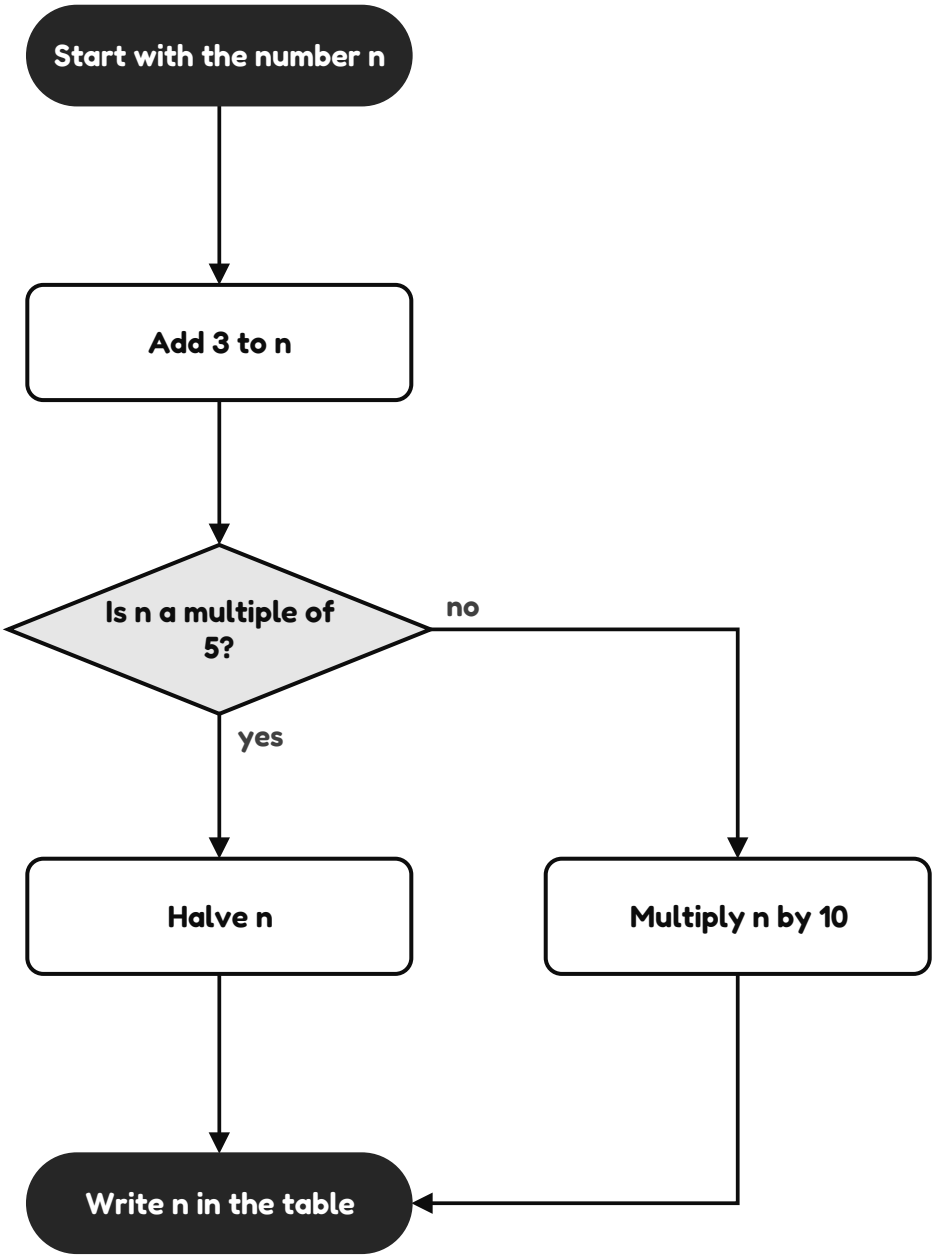
☆ Color the star when you finish!

Sample pages

Follow the Flowchart

★☆☆ **No. 3**

Follow the arrows, starting at the top, with each starting number **n** from the table. Do what each box says (n changes as you go) and answer each question **yes** or **no**. Write what comes out in the table.



n	Out
18	
1	
7	

Work space

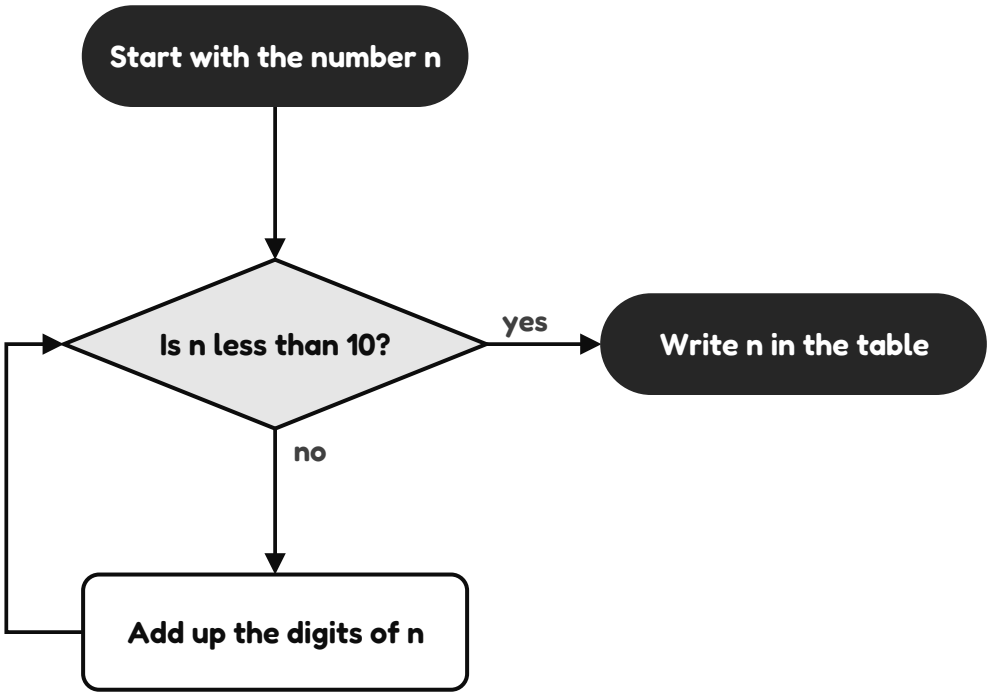
☆ Color the star when you finish!

Sample pages

Follow the Flowchart

★★☆ **No. 4**

Follow the arrows, starting at the top, with each starting number **n** from the table. Do what each box says (n changes as you go) and answer each question **yes** or **no**. Write what comes out in the table.



n	Out
687	
872	
937	

Work space

☆ Color the star when you finish!

Sample pages