Q: What is the value in x when the code is run?

iClicker

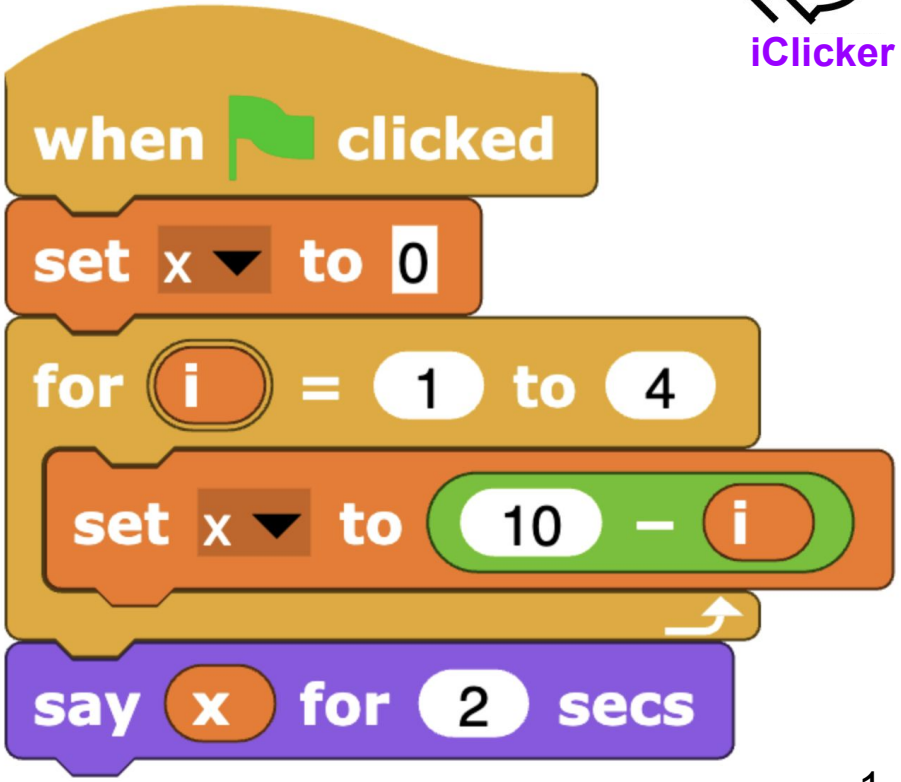
A. 1

B. 3

C. 4

D. 6

E. 10



CPSC 100

Computational Thinking

Mod Operator + Debugging

Instructor: Parsa Rajabi
Department of Computer Science
University of British Columbia

Agenda

- Learning Goals
- Course Admin
- Programming Concepts + Challenges
  - Mod Operator
  - Debugging

Learning Goals

After this today's lecture, you should be able to:

- Understand and explain the modulo (mod) operator.
- Apply mod operator in Snap! programming
- Understand the history & importance of debugging in programming
- Identify any bugs associated with a given code block
- Explain in plain English what needs to be changed to resolve bugs
- *Bonus: understand AM/PM acronym in the clock system*

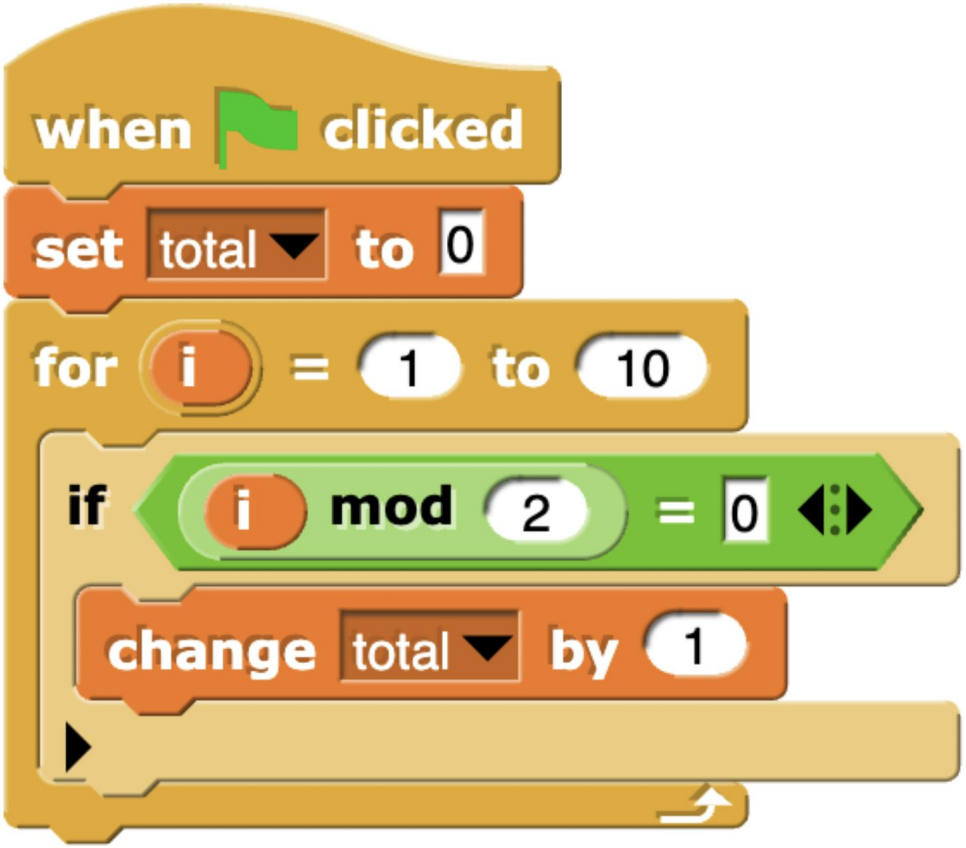
Course Admin

Course Admin

- **Post-Class (PC) Quiz #3**
  - Due on Sunday, Feb 9 at 11:59pm
- **Project Milestone 1**
  - Due on Wednesday, Feb 12 at 11:59pm
- **Midterm**
  - Review exam details here
  - 📅 February 14 💘
  - ⏰ 3:00-3:50 pm
  - ‼️⚠️ CHBE 101 [**Ch**emical and **B**iological **E**ngineering Building] ⚠️‼️
  - Practice Problems available here
  - Past Exams posted on Canvas

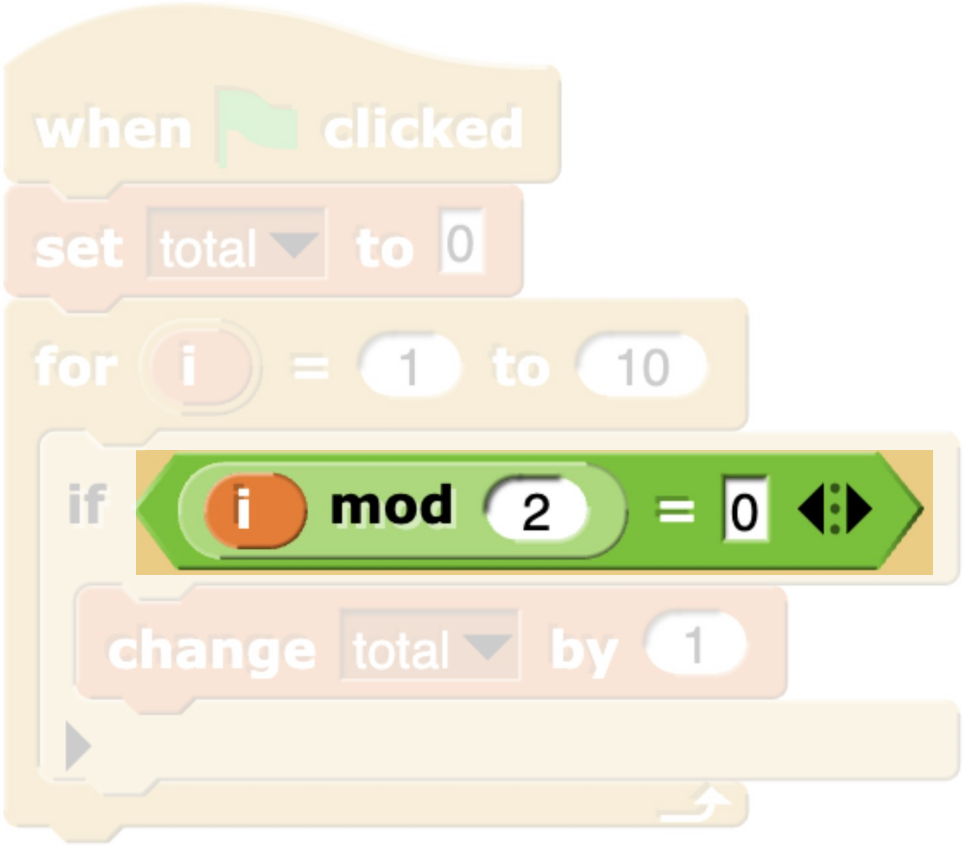
Q: What is the value of total when this code block is run?

What does this code block do?



"Mod" Operator

Short for *modulo*



$$25 \div 7$$

Image Source

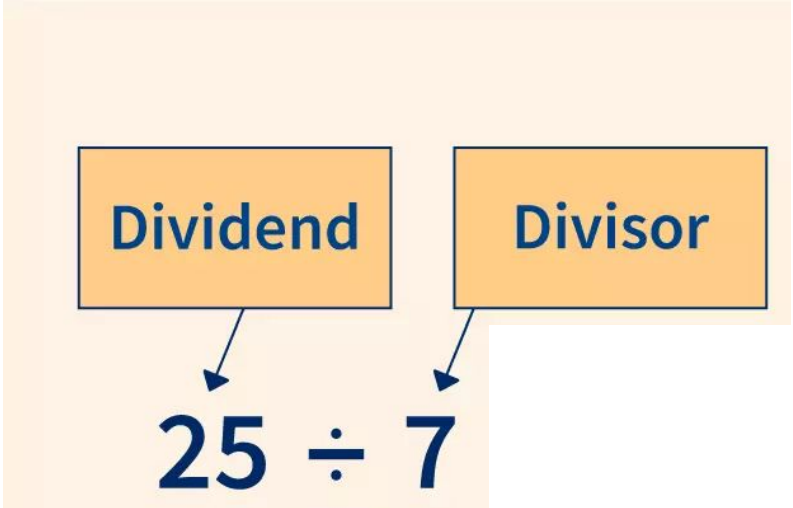


Image Source

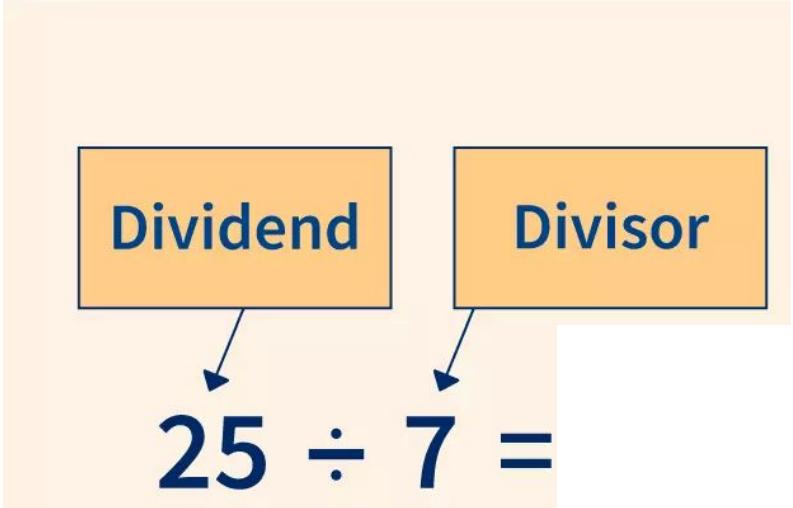


Image Source

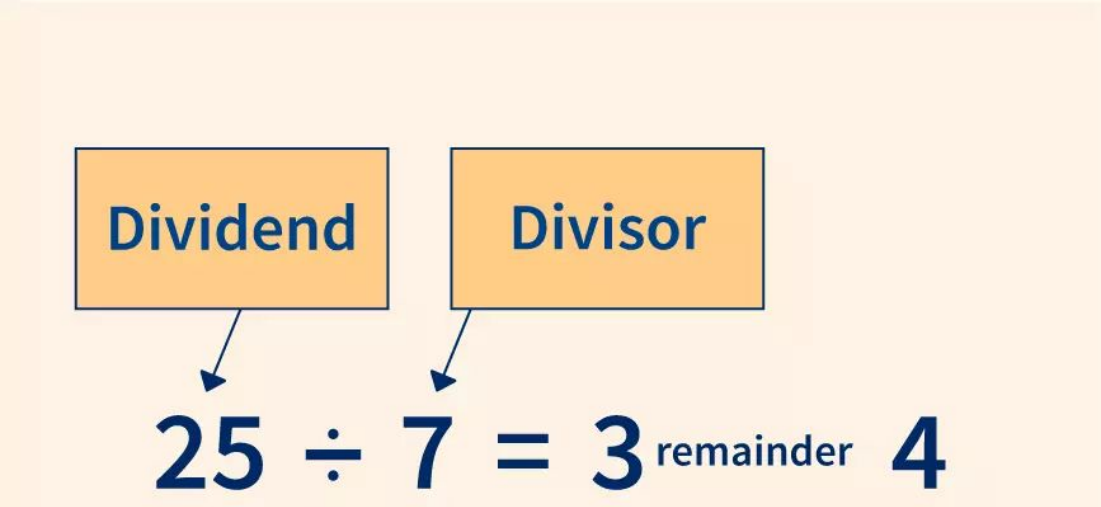


Image Source

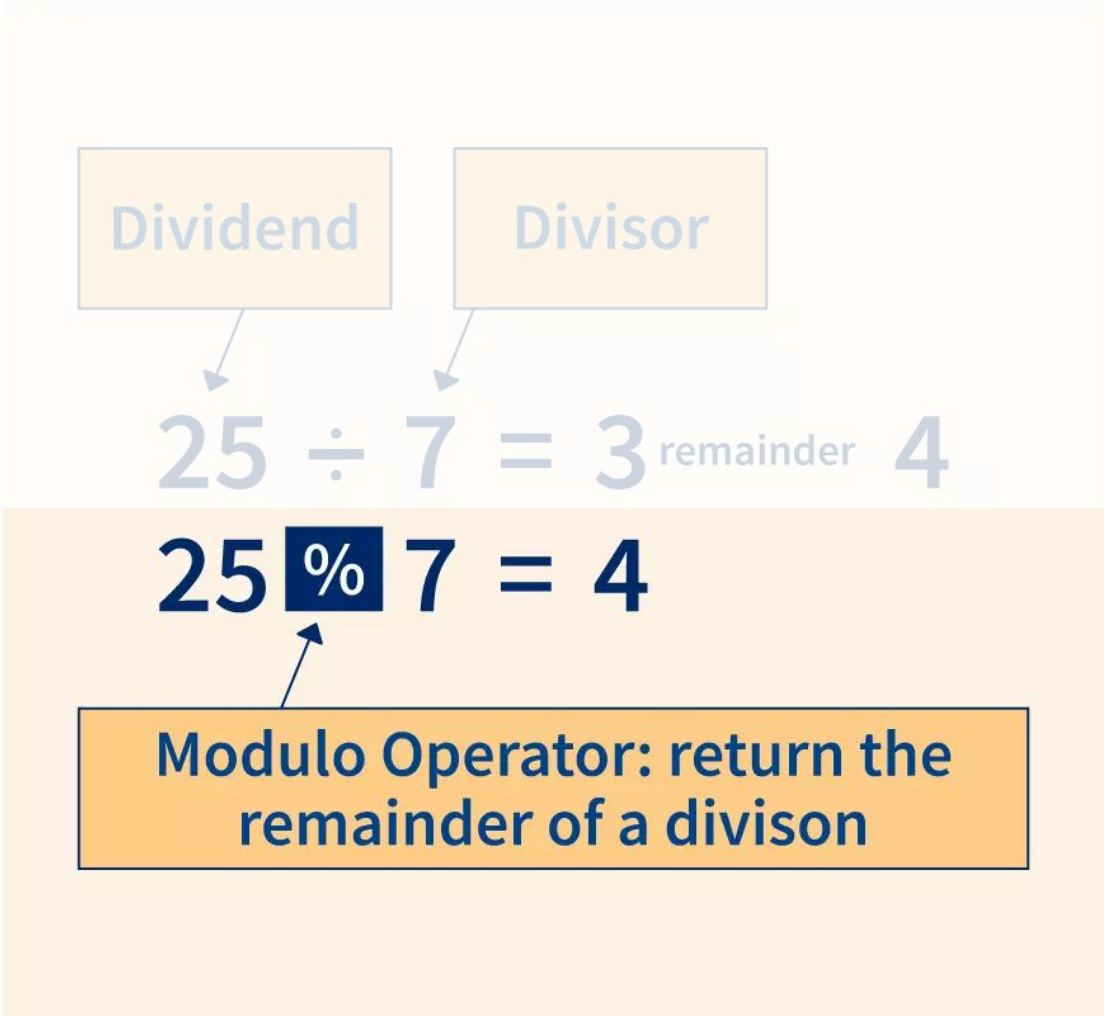


Image Source

Examples

- Clock System (AM/PM)
  - Before noon: **A**nte **M**eridiem (AM) → 12:01am to 11:59am
  - After noon: **P**ost **M**eridiem (PM) → 12:00pm to 11:59pm

Examples

- Clock System (AM/PM)
  - Before noon: **A**nte **M**eridiem (AM) → 12:01am to 11:59am
  - After noon: **P**ost **M**eridiem (PM) → 12:00pm to 11:59pm
- Clock System (military)
  - 24 hour system → after 12: we keep on counting (13, 14..)

Examples

- Clock System (AM/PM)
  - Before noon: **A**nte **M**eridiem (AM) → 12:01am to 11:59am
  - After noon: **P**ost **M**eridiem (PM) → 12:00pm to 11:59pm
- Clock System (military)
  - 24 hour system → after 12: we keep on counting (13, 14..)
- To convert between these two, we use a **mod operator!**
  - Our class starts at 15:00 → 3pm
  - 15 **mod** 12 = 3 *(since dividing 15 by 12, the remainder is 3)*

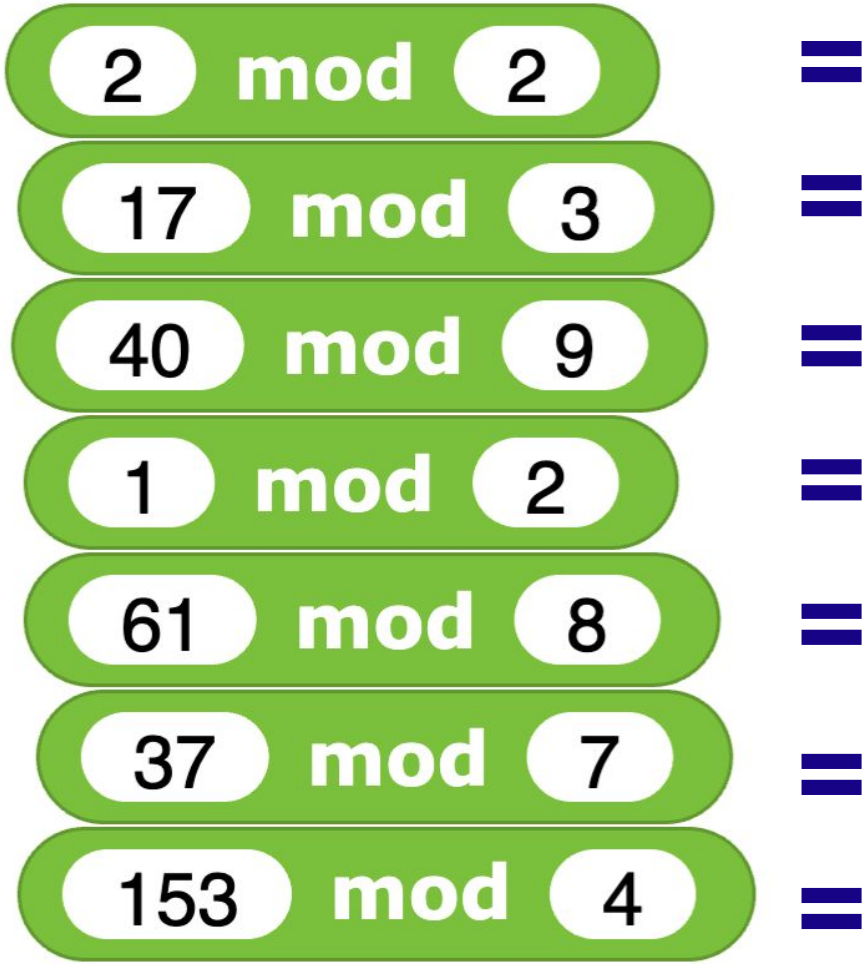
More Examples

- 5 mod 2 = 1 (the closest divisor is [2], 2x2 = **4**, the remainder is 1)
- 9 mod 3 = 0 (since 9 is exactly divisible by 3 with **no** remainder)
- 17 mod 5 = 2 (the closest divisor is [3], 5x3 = **15**, the remainder is 2)

Take-Home Practice

More Examples

- 5 mod 2 = 1 (the closest divisor is [2], 2x2 = **4**, the remainder is 1)
- 9 mod 3 = 0 (since 9 is exactly divisible by 3 with **no** remainder)
- 17 mod 5 = 2 (the closest divisor is [3], 5x3 = **15**, the remainder is 2)
- 25 mod 3 = 1 (the closest divisor is [8], 3x8 = **24**, the remainder is 1)
- 44 mod 10 = 4 (the closest divisor is [4], 10x4 = **40**, the remainder is 4)
- 53 mod 6 = 5 (the closest divisor is [8], 6x8 = **48**, the remainder is 5)
- 72 mod 8 = 0 (since 72 is exactly divisible by 8 with **no** remainder)

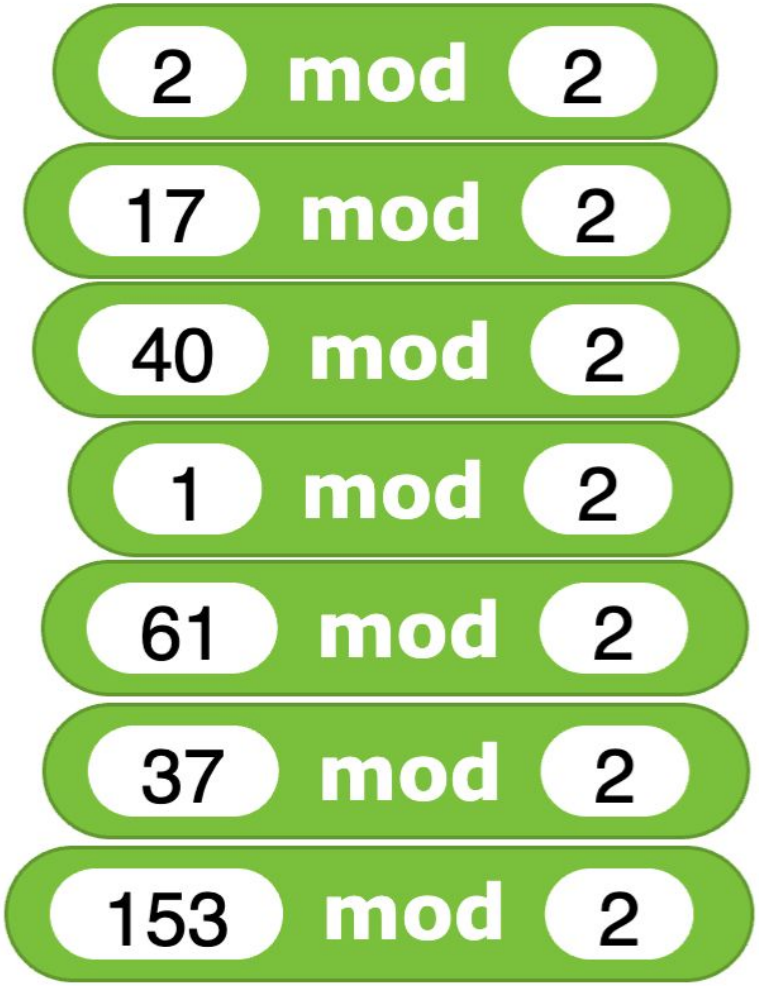

2 mod 2 =
17 mod 3 =
40 mod 9 =
1 mod 2 =
61 mod 8 =
37 mod 7 =
153 mod 4 =

Programming Context:

- This operator is helpful in programming to check **if a number is even or odd**, **looping through a range of values**, and **creating patterns**.

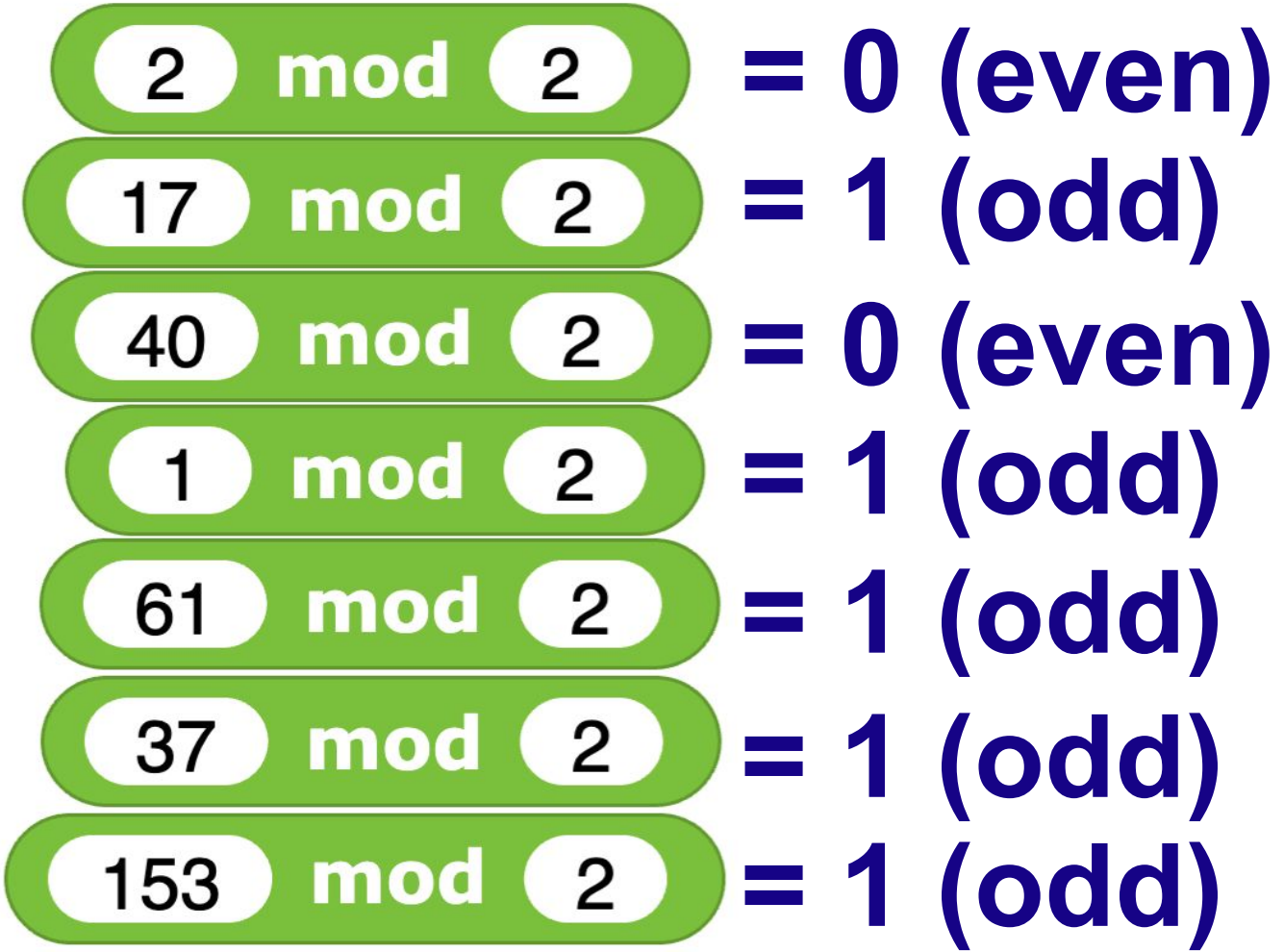
Programming Context:

- This operator is helpful in programming to check **if a number is even or odd**, **looping through a range of values**, and **creating patterns**.
- An even number will have a remainder of 0 **when divided by 2**, while an odd number will have a remainder of 1
  - 7 mod 2 = 1 (Odd)
  - 12 mod 2 = 0 (Even)

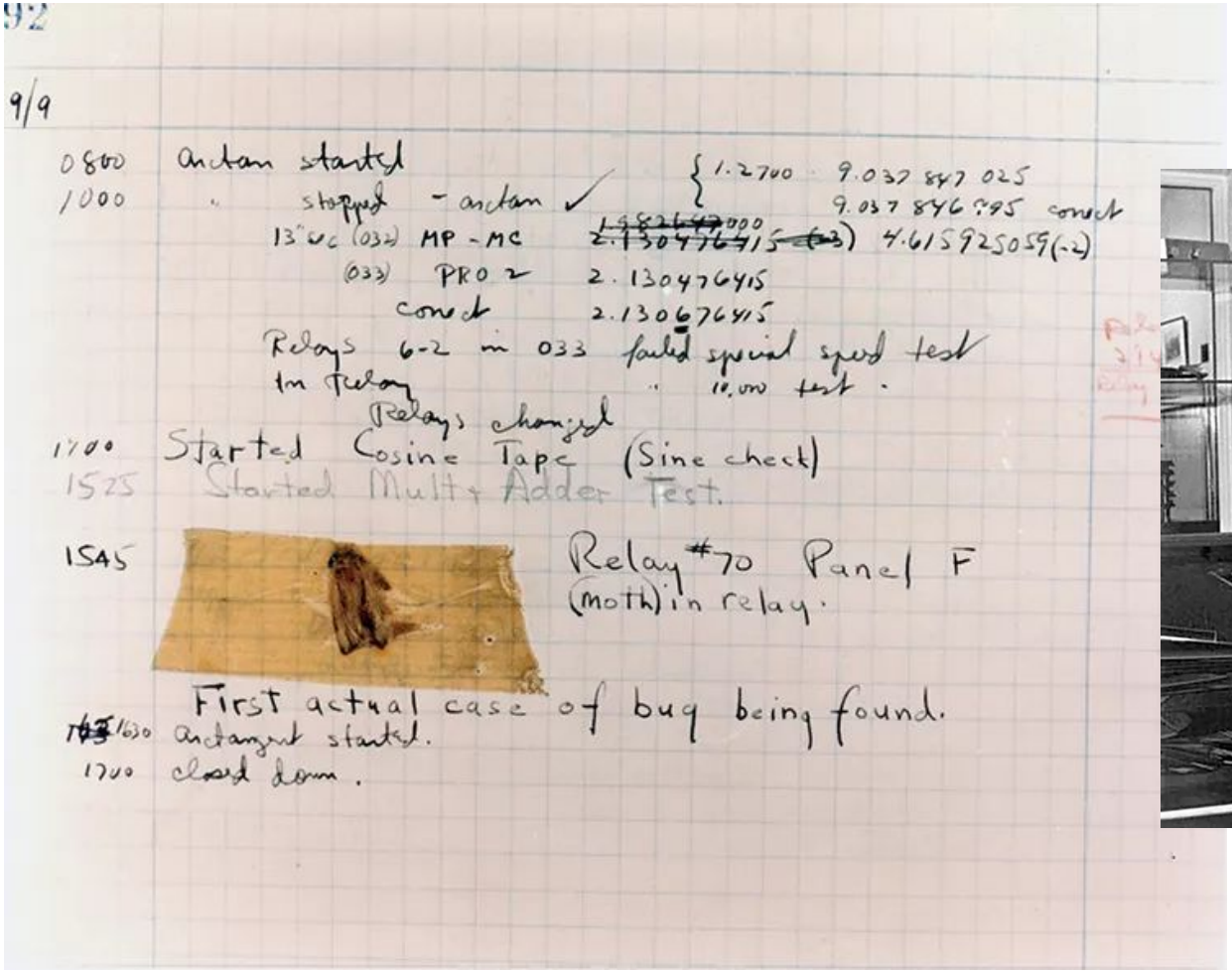


What do these Arithmetic Operators evaluate to?

Odd or even?


2 mod 2 = 0 (even)
17 mod 2 = 1 (odd)
40 mod 2 = 0 (even)
1 mod 2 = 1 (odd)
61 mod 2 = 1 (odd)
37 mod 2 = 1 (odd)
153 mod 2 = 1 (odd)

Debugging



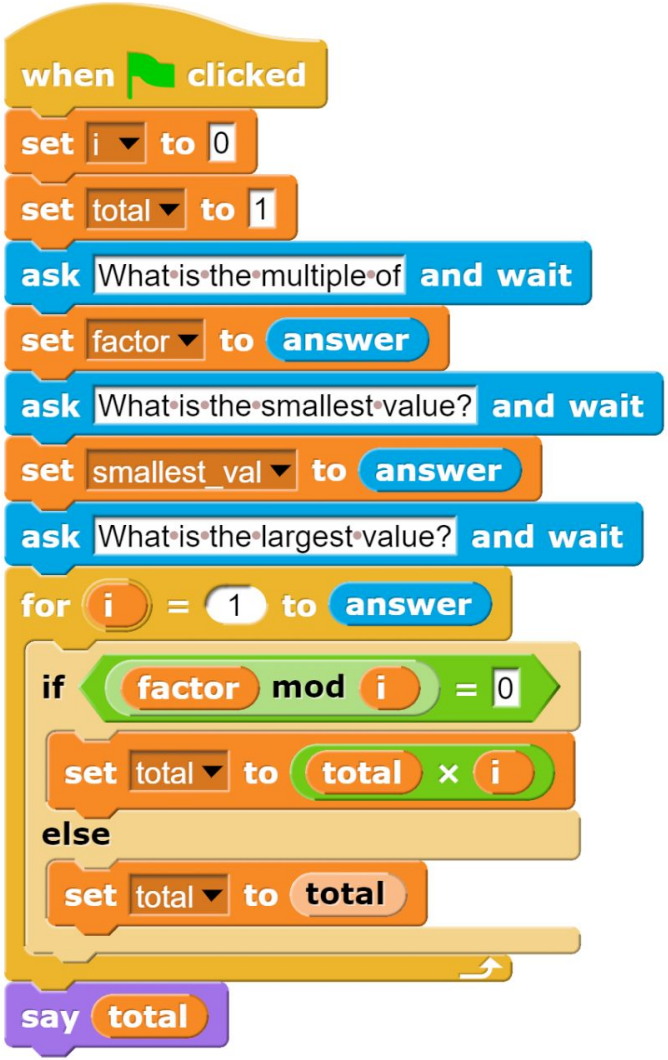
92

9/9

0800 Arctan started
1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 795 correct
13" vc (032) MP - MC ~~1.9821447000~~ 2.13047649 15 (-2) 4.615925059(-2)
(033) PRO 2 2.13047649 15
correct 2.13067649 15
Relays 6-2 in 033 failed special speed test
In relay 10.00 test.
Relays changed
1700 Started Cosine Tape (Sine check)
1525 Started Multy Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
~~1700~~ 1630 Arctangent started.
1700 closed down.

Reference: The "bug" [Link]

Activity



This code block is supposed to find the product between two positive integers (not inclusive)

Example, if the user inputs:

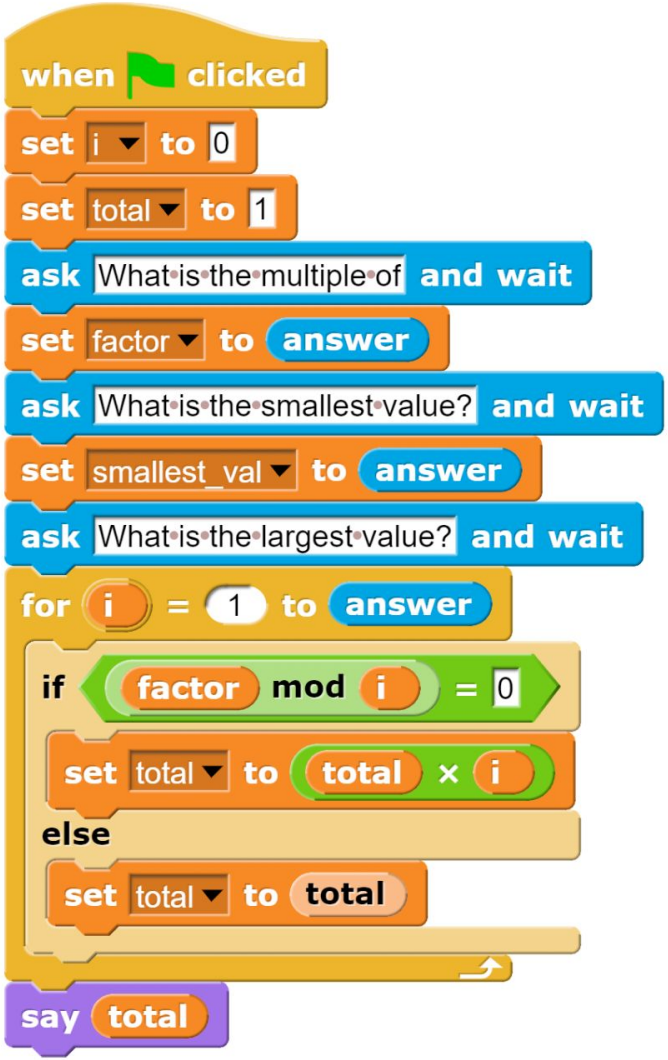
What is the multiple of? **2**

What is the smallest value? **1**

What is the largest value? **10**

The result should be:

2x4x6x8 = 384



Review the code block and identify any bug(s).

1. Clearly highlight the problematic code [bug(s)]
2. Explain in plain English what needs to be changed so the code works properly

Demo Solution

Wrap up

Wrap Up

- **Post-Class (PC) Quiz #3**
  - Due on Sunday, Feb 9 at 11:59pm
- **Project Milestone 1**
  - Due on Wednesday, Feb 12 at 11:59pm
- **Midterm**
  - Review exam details here
  - 📅 February 14 💘
  - ⏰ 3:00-3:50 pm
  - ‼️⚠️ CHBE 101 [**Ch**emical and **B**iological **E**ngineering Building] ⚠️‼️
  - Practice Problems available here
  - Past Exams posted on Canvas