Project 5: Climbing a Silly Tower

COSC 1423 | Fall 2023
Instructor: Megan Avery

Topics: while loops, input validation, logical operators

Goal: Simulate climbing to the top of a tower that has some silly bricks built in. The silly bricks make you slide back down the tower.

Turn In Instructions:
A zip file called project05_firstname_lastname.zip containing the following files:
Implementation: Your python file named proejct05_firstname_lastname.py.
⚠️ Documentation: ⚠️
A pdf called project05_firstname_last.pdf that contains your objective definition, flowchart, test cases, and followup questions.

Objective Definition (15 pts): See Planning Guide

Style (10 pts): See Style Guide

Implementation (55 pts):

- Background
  - A climber is climbing a tower of some height, measured in bricks. The tower is made up of 3 types of bricks. For the 1st type, rock, nothing special happens when it is hit. The other 2 types of bricks are strategically placed and cause the climber to slide back down some number of bricks.
  - You are in charge of setting the following specs:
    - The height of the tower
    - The formulas for both silly bricks and how far down a user slides for each.
      - An example would be a slime brick when the brick number ends in a 7 and they slide down 7 bricks. You may **not** use this example in your program.
      - *Hint: be careful how you set these so an infinite loop doesn't happen*
  - The climber starts on the ground at the beginning of every run of the program.
- Specifics
  - The climber will climb the tower until they reach the top. Every round print the climber's current position and ask the climber how far they would like to climb, which must be between 1 and 15 bricks inclusive. If the climber lands on one of your silly bricks, print a message about backsliding and update their height accordingly. After the climber has reached the top, print how many climbs they made.

Test Cases (10 pts): See Project FAQ

Followup Questions (10 pts):

1. How long did this project take you?
2. Did you complete the extension?
3. How tall is your tower and what are your 2 rules for silly bricks?
4. Explain how you did the backsliding logic.
5. Explain your usage of AI in this project. What type of queries did you do during development?

Extension (+5 pts):
Print a message about the longest valid mini climb the user made during their ascent.

Sample Run (slime brick when height ends in a 7):

```
Tower height: 50

Current height: 0
How many do you want to climb this round? (whole feet): 12

Current height: 12
How many do you want to climb this round? (whole feet): 15
😿 Oops! You hit a slime brick and slid back down 7 feet!

Current height: 20
How many do you want to climb this round? (whole feet): 90
😡 Not a valid climb, try again: -23
😡 Not a valid climb, try again: 5

Current height: 25
How many do you want to climb this round? (whole feet): 14

Current height: 39
How many do you want to climb this round? (whole feet): 15

You used 5 mini climbs to climb the tower!
```