EPFL CIVIL-127, Lab 2 solution

2.1

```python
# Simulate lots of Snakes & Ladders games and calculate average
# number of moves per game.

import random

sum = 0
N = 1_000_000
for i in range(N):
    moves = 0
    marker = 1
    while marker != 100:
        # roll the dice and update marker
        dice = random.randint(1, 6)
        marker += dice
        moves += 1

        # handle bounce at 100
        if marker > 100:
            overflow = marker - 100
            marker = 100 - overflow

        # handle snakes and ladders
        if marker == 43:
            marker = 17
        elif marker == 50:
            marker = 5
        elif marker == 56:
            marker = 8
        elif marker == 73:
            marker = 15
        elif marker == 84:
            marker = 63
        elif marker == 87:
            marker = 49
        elif marker == 98:
            marker = 40
        elif marker == 2:
```

```
            marker = 23
        elif marker == 6:
            marker = 45
        elif marker == 20:
            marker = 59
        elif marker == 52:
            marker = 72
        elif marker == 57:
            marker = 96
        elif marker == 71:
            marker = 92

    sum += moves

average = sum / N
print("average:", average)
```

The value we are looking for is 84.853, anything in the 84-85 range can be considered acceptable.

2.2

- We take the code from the previous week.
- (L1-4) We update the top-level comment. Don't forget to update comments when you are updating code; out-of-date comments are a very common problem!
- (L8) We use `random.randint(1, 100)` to pick a number randomly
- (L13) We fix the bug with string comparisons using `int()`.
- (L15-21) We *refactor* the last few lines to use `if`, `elif`, and `else` to illustrate what we learned in class.

  Note: Refactoring is the process of restructuring existing code to improve its internal design and readability without altering its functionality.

```
1  # One-player number guessing game.
2  # Computer picks a number.
3  # Player keeps guessing. The program tells them if
4  # their number is too small, too big, or correct.
5
6  import random
7
8  secret = random.randint(1, 100)
9
10 attempts = 0
11 done = False
12 while not done:
13     guess = int(input("Enter your guess: "))
14     attempts += 1
15     if guess < secret:
16         print("Too small, try again")
17     elif guess > secret:
18         print("Too big, try again")
19     else:
20         print("You got it! Your score:", attempts)
21         done = True
```