CS50's Introduction to Programming with Python

OpenCourseWare

Donate (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan)
(https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)
(https://www.reddit.com/user/davidjmalan)
(https://www.threads.net/@davidjmalan) (https://twitter.com/davidjmalan)

Little Professor

One of David's first toys as a child, funny enough, was Little Professor (https://en.wikipedia.org/wiki/Little_Professor), a "calculator" that would generate ten different math problems for David to solve. For instance, if the toy were to display `4 + 0 =`, David would (hopefully) answer with `4`. If the toy were to display `4 + 1 =`, David would (hopefully) answer with `5`. If David were to answer incorrectly, the toy would display `EEE`. And after three incorrect answers for the same problem, the toy would simply display the correct answer (e.g., `4 + 0 = 4` or `4 + 1 = 5`).

In a file called `professor.py`, implement a program that:

- Prompts the user for a level, n. If the user does not input `1`, `2`, or `3`, the program should prompt again.
- Randomly generates ten (10) math problems formatted as `X + Y =`, wherein each of `X` and `Y` is a non-negative integer with n digits. No need to support operations other than addition (`+`).
- Prompts the user to solve each of those problems. If an answer is not correct (or not even a number), the program should output `EEE` and prompt the user again, allowing the user up to three tries in total for that problem. If the user has still not answered correctly after three tries, the program should output the correct answer.
- The program should ultimately output the user's score: the number of correct answers out of 10.

Structure your program as follows, wherein `get_level` prompts (and, if need be, re-prompts) the user for a level and returns `1`, `2`, or `3`, and `generate_integer` returns a randomly generated non-negative integer with `level` digits or raises a `ValueError` if `level` is not `1`, `2`, or `3`:

```
import random


def main():
    ...


def get_level():
    ...


def generate_integer(level):
    ...


if __name__ == "__main__":
    main()
```

▼ **Hints**

- Note that you can raise an exception like `ValueError` with code like:

  ```
  raise ValueError
  ```

- Note that the `random` module comes with quite a few functions, per docs.python.org/3/library/random.html (https://docs.python.org/3/library/random.html).

Demo

Recorded with asciinema

Before You Begin

Log into cs50.dev (https://cs50.dev/), click on your terminal window, and execute `cd` by itself. You should find that your terminal window's prompt resembles the below:

```
$
```

Next execute

```
mkdir professor
```

to make a folder called `professor` in your codespace.

Then execute

```
cd professor
```

to change directories into that folder. You should now see your terminal prompt as `professor/ $`. You can now execute

```
code professor.py
```

to make a file called `professor.py` where you'll write your program.

How to Test

Here's how to test your code manually:

- Run your program with `python professor.py`. Type `-1` and press Enter. Your program should reprompt you:

  ```
  Level:
  ```

- Run your program with `python professor.py`. Type `4` and press Enter. Your program should reprompt you:

  ```
  Level:
  ```

- Run your program with `python professor.py`. Type `1` and press Enter. Your program should begin posing addition problems with positive, single-digit integers. For example:

  ```
  6 + 6 =
  ```

  Your program should output 10 distinct problems before printing the number of questions you answered correctly and exiting.

- Run your program with `python professor.py`. Type `1` and press Enter. Answer the first question incorrectly. Your program should output:

  ```
  EEE
  ```

  before reprompting you with the same question.

- Run your program with `python professor.py`. Type `1` and press Enter. Answer the first question incorrectly, three times. Your program should output the correct answer. For example:

  ```
  6 + 6 = 12
  ```

  and then move on to another question. Answer the remaining questions correctly. Your program should output a score of `9`.

- Run your program with `python professor.py`. Type `1` and press Enter. Answer all 10 questions correctly. Your program should output a score of `10`.

You can execute the below to check your code using `check50`, a program that CS50 will use to test your code when you submit. But be sure to test it yourself as well!

```
check50 cs50/problems/2022/python/professor
```

Green smilies mean your program has passed a test! Red frownies will indicate your program output something unexpected. Visit the URL that `check50` outputs to see the input `check50` handed to your program, what output it expected, and what output your program actually gave.

How to Submit

In your terminal, execute the below to submit your work.

```
submit50 cs50/problems/2022/python/professor
```