MTH 3300
Summer 2024

Programming Problem 2b: binary.py

Directions: Download the template files I have provided on Blackboard. Then open Spyder, load these template files, and write the following programs. Submit your source code via Gradescope in `.py` format. READ THE INSTRUCTIONS on submitting your work in the Course Documents section of Blackboard.

Specify collaborators/resources or explicitly specify that none were used in the comments at the top of your `.py` file. Failure to include this will result in a zero on the assignment.

Be sure to read the SPECIFICATIONS carefully! And write comments!

Recall that we briefly discussed the *binary system* (or *base two*) in class. A number is represented in binary by a sequence of `1`'s and `0`'s (also known as *bits*), which have place values given by powers of 2 instead of powers of 10. In an eight-digit binary number, the left digit would be the 128's digit, the next digit would be the 64's digit, the third would be the 32's digit, followed by the 16's, 8's, 4's, 2's, and 1's digits.

For example, `01011101` in binary would represent the (base-10) number 93, because this number has, reading from the left, no 128, one 64, no 32, one 16, one 8, one 4, no 2, and one 1, and 64 + 16 + 8 + 4 + 1 = 93. And `00000111` would represent the (base-10) number 7, since this number has no 128, no 64, no 32, no 16, no 8, one 4, one 2, and one 1.

Write a program that takes **no input from the user**, but prints a random 4-digit binary number with 4 decimals, and its base-10 equivalent. Specifically, your program should use the `random` module to pick 8 random numbers that should each be either `0` or `1`. The program should then display those 8 digits in a row in the format `xxxx.xxxx`, together with the equivalent base 10 number.

When you run your program, a sample run might look like this:

```
Here's a random example of binary!
The binary number 1 0 0 0 . 1 1 1 0 is the same as the decimal number 8.875.
```

or this:

```
Here's a random example of binary!
The binary number 0 0 1 1. 0 0 1 1 is the same as the decimal number 3.1875.
```

Every time I run the program, I should get a different number.

You should be able to complete this program (and the challenge) only using things we have discussed already in the class: **don't** use lists or tuples (these are sequences of variables enclosed in either `[]` or `()`), **don't** use loops, and definitely **don't** use any of Python's functions for binary (like `bin`). If/else statements are acceptable, but unnecessary. Instead, use mathematics – multiplication, exponentiation, and maybe `//` and `%` (the last two might helpful for the challenge).

Specifications: your program must

- accept NO user input.
- use `random` to generate eight different random numbers – each should be 0 or 1.
- display the eight generated digits, as well as the equivalent decimal number, in the manner shown above. In particular, I want to see binary numbers listed like `1 0 1 0 .  1 1 0 0`, *not* like `(1, 0,`

`1, 0, 1, 1, 0, 0)`. (`1010.1100`, without the spaces in between, *is* acceptable.)

- display different numbers each time the program is run.
- only use techniques that we have discussed in class thus far – in particular, **no use of the `bin` function, no use of loops, and no use of lists or tuples**.