

## Problem 1. Input loop

Write a program that repeatedly asks the user for names, and then outputs the list of all the names before the special name END is entered. An example of an interaction would be:

```
Enter a name: Alice
Enter a name: Bob
Enter a name: Charlie
Enter a name: Dave
Enter a name: Emily
Enter a name: END
The names are: Alice, Bob, Charlie, Dave, Emily
```

To store the names in the program, you can use a string. For example, the string might be "" at first, then "Alice", then "Alice, Bob", *etc.* You can use `.extend` to add a name to the string.

## Problem 2. Sum more $\pi$

Now we would like to figure out how many terms in the summation above we need to add up in order to approximate  $\pi$  to  $n$  significant digits. Write a function that returns the number of the terms required

to obtain an approximation of  $\pi$  using the Leibniz formula that agrees with the actual value of  $\pi$  to  $n$  significant digits (for the purposes of this problem, the approximation and  $\pi$  agree to  $n$  significant digits if the first  $n$  digits are the same in  $\pi$  and in the approximation.)

The best approximation of  $\pi$  using a `float` is available in `math.pi` (execute `import math` to be able to use it.)

Part of your job is to figure out whether two numbers agree to  $n$  significant digits. To figure that out, for a float `x`, consider what `round(x*(10**n))` means (try `round(math.pi*(10**5))` to see what it does).

## Problem 3. Pseudo-random numbers

Here is how we generated pseudo-random numbers in class:

```
def initialize():
    global a, b, c, cur, my_max
    a = 438750932854830938457
    b = 287492837491
    cur = 1
    my_max = 100000
```

```
def next_cur():
    global cur
    cur = (cur * a % b)

    return cur % my_max
```

```
def myrandom():
```

```
    return next_cur()/my_max

initialize()
if __name__ == '__main__':

    print(myrandom())
    print(myrandom())
```

### Part (a)

Set up two files, `myrandom.py` and `test_myrandom.py`. Put the code above in `myrandom.py`. In `test_myrandom.py`, write a test function that prints the output for several calls to `myrandom()` .

To do that, put

```
import myrandom
if __name__ == '__main__':
    print(myrandom.myrandom())
    print(myrandom.myrandom())
```

in the file `test_myrandom.py`, place both `myrandom.py` and `test_myrandom.py` in the same folder, and run `test_myrandom.py` using Run > Run File As Script.

### Part (b)

Here, we will investigate why `a` and `b` were chosen to be what they were. Modify `myrandom.py` so that `a` is 10 and `b` is 7. You will notice that the output now cycles.

Write a function that repeatedly calls `myrandom()` in order to find out and return the *length* of the cycle of the pseudo-random number when starting from a given initial seed `cur`, `a`, and `b`. To avoid very long loops, you can stop after 10000 iterations and return -1 if the cycle length is more than 10000.

Find values of `a` and `b` for which the cycle length is longer than 10000.

## Problem 4.

Work on Project 1.