

3. Python: Log Decorator

**ALL**

Implement a decorator that logs the invocation of the decorated function to the provided file descriptor. You can assume that it will only decorate functions that take in positional arguments.

- The log line should follow this format **LOG: <function_name>** (**<comma_separated_call_parameters>**). The last character should be a newline.
- The line should be logged prior to the execution of the decorated function's body.

Example

```
@log(descriptor)
def my_max(a, b, c):
    return max(a, b, c)
```

```
@log(descriptor)
def my_min(a, b):
    return min(a, b)
```

```
my_max(20, 9, 10)
my_min(1, 2)
```

Writes to the given *descriptor* the following output

```
LOG: my_max(20,9,10)
LOG: my_min(1,2)
```

Your implementation of the decorator will be tested by a provided code stub on several input files. Each input file contains parameters to call a decorated function. The provided code calls the functions and writes the returned values to the descriptor.

Constraints:

The decorated function takes only positional arguments.

► Input Format Format for Custom Testing

▼ Sample Case 0

Sample Input 0