

junhoc_A7

November 26, 2018

0.0.1 Problem 1

Problem 1 of Chapter 7

In [1]: *## small.py*

```
def smallest_factor(n):  
    """  
        Return the smallest prime factor of  
        the positive integer n.  
    """  
    if n == 1: return 1  
    for i in range(2, int(n ** 0.5)):  
        if n % i == 0: return i  
    return n
```

In [3]: *## test_small.py*

```
## from small import smallest_factor  
  
def test_smallest_factor():  
    assert smallest_factor(25) == 5, "Failed on squares of primes"
```

Problem 2 of Chapter 7

In []: *## monlen.py*

```
def month_length(month, leap_year=False):  
    """Return the number of days in the given month."""  
    if month in {"September", "April", "June", "November"}:  
        return 30  
    elif month in {"January", "March", "May", "July",  
                   "August", "October", "December"}:  
        return 31  
    elif month == "February":  
        if not leap_year:  
            return 28  
        else:
```

```

        return 29
    else:
        return None

```

In []: *## test_monlen.py*

```

## from monlen import month_length

```

```

def test_month_length():

    month_day_dict = {
        "January": 31,
        "February": {"leap": 29, "not_leap": 28},
        "March": 31,
        "April": 30,
        "May": 31,
        "June": 30,
        "July": 31,
        "August": 31,
        "September": 30,
        "October": 31,
        "November": 30,
        "December": 31
    }

    error_msg_month = "Month not matching day!"
    error_msg_february = "Month not matching day, and " + \
        "check leap year status!"
    error_msg_sthelse = "If input is not the name of a " + \
        "month, then should return None!"

    for key, value in month_day_dict.items():
        if key == "February":
            assert month_length(key, True) == \
                month_day_dict[key]["leap"], error_msg_february
            assert month_length(key, False) == \
                month_day_dict[key]["not_leap"], error_msg_february
        else:
            assert month_length(key) == \
                month_day_dict[key]
    assert month_length("Something_Else") == None, error_msg_sthelse

```

Problem 3 of Chapter 7

In []: *## oper.py*

```

def operate(a, b, oper):
    """Apply an arithmetic operation to a and b."""

```

```

if type(oper) is not str:
    raise TypeError("oper must be a string")
elif oper == '+':
    return a + b
elif oper == '-':
    return a - b
elif oper == '*':
    return a * b
elif oper == '/':
    if b == 0:
        raise ZeroDivisionError("division by zero is " + \
                                "undefined")
    return a / b
raise ValueError("oper must be one of '+', " + \
                 "'/', '-', or '*'")

```

In []: *## test_oper.py*

```

## from oper import operate
import pytest

```

```
def test_operate():
```

```
    ## Testing additions
```

```
    assert operate(4, 2, '+') == 6, "integer addition"
    assert operate(4, 0.25, '+') == 4.25, "float addition"

```

```
    ## Testing subtractions
```

```
    assert operate(4, 5, '-') == -1, "integer subtraction"
    assert operate(4, 3.75, '-') == 0.25, "float subtraction"

```

```
    ## Testing multiplications
```

```
    assert operate(4, 5, '*') == 20, "integer multiplication"
    assert operate(4, 0.1, '*') == 0.4, "float multiplication"

```

```
    ## Testing divisions
```

```
    assert operate(4, 2, "/") == 2, "integer division"
    assert operate(0.06, 0.3, "/") == 0.2, "float division"

```

```
    ## Testing exceptions
```

```
    with pytest.raises(ZeroDivisionError) as excinfo0:
        operate(4, 0, "/")
    assert excinfo0.value.args[0] == "division by zero is " + \
                                     "undefined"

```

```
    with pytest.raises(ValueError) as excinfo1:
```

```
        operate(4, 0, "string")
    assert excinfo1.value.args[0] == "oper must be one of '+', " + \

```

```
        "'/', '-', or '*'"

with pytest.raises(TypeError) as excinfo2:
    operate(4, 0, 1)
assert excinfo2.value.args[0] == "oper must be a string"
```