Designing Functions

Introduction to Computer Programming

Dr. Paul Vrbik

September 19, 2018

Question

1. Each adult eats 3 cookies,
2. each child eats 2, and
3. each teen eats 5.

Write a function that takes three parameters representing the number of adults, children, and teens, and returns the required number of cookies.

```
def cookies_required(adults:int, children:int, teens:int) -> int:
    """(int, int, int) -> int
    Returns number of cookies required for adults-many
    adults, children-many children, and teen-many teens.
    >>> cookies(0, 0, 0)
    0
    >>> cookies(2, 3, 4)
    32
    """
    return 3*adults + 2*children + 5*teens
```

sandbox.py

```
]$ python -i sandbox.py
>>> vrbik = 2
>>> dir()
[‘__annotations__’, ‘__builtins__’, ‘__cached__’,
 ‘__doc__’, ‘__loader__’, ‘__name__’, ‘__package__’,
  ‘__spec__’, ‘cookies_required’, ‘vrbik’]
```

`dir()` *returns the list of names in the current local scope.*

```
>>> help(cookies_required)
Help on function cookies_required in module __main__:

cookies_required(adults, children, teens)
    Return number of cookies required for adults-many
    adults, children-many children, and teens-many teens.
    >>> cookies(0, 0, 0)
    0
    >>> cookies(2, 3, 4)
    32
```

Question

A student is eligible for PEY if their CGPA is at least 2, they are in year 2 or year 3, and they are in a CS program.

Write a function that determines if a student is eligible for PEY.

```
def is_PEY_eligible(cgpa:float, year:int, in_cs:bool) -> bool:
    """(float, int, bool) -> bool
    Returns true when PEY eligibility is satisfied.
    >>> is_PEY_eligible(2, 3, True)
    True
    >>> is_PEY_eligible(1, 3, True)
    False
    """
    return (cgpa >= 2) and (year == 2 or year == 3) and in_cs
```

Question

Write a function that returns the volume of a sphere given the sphere's radius.

```
PI = 3.14159

def sphere_volume(radius:float) -> float:
    """(float) -> float
    Returns the volume of a sphere given radius.
    >>> sphere_volume(0)
    0
    >>> sphere_volume(0.5)
    0.5235983333333333
    """
    return 4/3*PI*radius**3
```

Question

1. Adults order two slices,
2. teens order three slices, and
3. children order one slice.

Each pizza has eight slices. Write a function that takes three parameters representing the number of adults, teens, and children, and returns the required number of pizzas.

```python
def required_pizza(adults:int, teens:int, children:int) -> int:
    """(int, int, int) -> int
    Returns the required number of 8-slice pizzas given
    the number of adults, teens, and children who eat 2,
    3, and 1 slice respectively.
    >>> required_pizza(0, 1, 2)
    1
    >>> required_pizza(4, 3, 1)
    3
    """
    slices = 2*adults + 3*teens + 1*children
    return slices//8 + (slices % 8 != 0)
```

Question

Write a function that takes three lengths and determines if there is a right-triangle with those sides.

```
def is_right_triangle(x:float, y:float, z:float) -> bool:
    """(float, float, float) -> bool
    Returns true only when there is a triangle with side lengths
    x, y, and z that is a right triangle.
    >>> is_right_triangle(3, 4, 5)
    True
    >>> is_right_triangle(0, 0, 1)
    False
    """
    A = x**2 - y**2 - z**2 == 0
    B = y**2 - z**2 - x**2 == 0
    C = z**2 - x**2 - y**2 == 0
    return A or B or C
```

Question

Write a function that returns the sum of the first n positive numbers — do not use loops.

```
def sum_of_first_n(n:int) -> int:
    """(int) -> int
    Returns the sum of the first n integers
    >>> sum_of_first_n(10)
    51
    >>> sum_of_first_n(0)
    0
    """
    return int(n*(n+1)/2)
```

Question

Write a function that determines if a student's `volume(float, float, float) -> float` function is correct by verifying three examples.

```
def check_student_answer(student_submission:function) -> bool:
    """(int) -> int
    Returns True only when student answer satisfies test case.
    >>> check_student_answer(lambda x,y,z : x*y*z)
    True
    >>> check_student_answer(lambda 0)
    False
    """
    Test01 = student_submission(1,2,3) == 6
    Test02 = student_submission(4,9,1) == 36
    Test03 = student_submission(0,9,1) == 0
    return Test01 and Test02 and Test03
```

Question

Write a function that takes two (unary) functions with signature `int -> int` and returns true function which is larger for some specified input x.

(Requires the `if` statement.)

```
def larger_unary(f:function, g:function, x:int) -> function:
    """(int) -> int
    Returns f when f(x) > g(x) and g otherwise.
    >>> def f(x):
             x**2
    >>> def g(x):
             2*x
    >>> larger_unary(f, g, 3)
    <function f at 0x... >
    >>> larger_unary(f, g, 2)
    <function g at 0x... >
    """
    return f if f(x) > g(x) else g
```

Question

Write a function that prints “twinkle, twinkle, little star ... ”

```
def print_poem() -> None:
    """() -> None
    Prints the poem ‘‘twinkle, twinkle, little star’’
    """
    print(" Twinkle, twinkle, little star,\n How I wonder what you
    print(" When this blazing sun is gone,\n When he nothing shines
    print(" Then the traveller in the dark\n Thanks you for your ti
    print(" In the dark blue sky you keep,\n And often through my c
    print(" As your bright and tiny spark\n Lights the traveller in
    return
```

Question

Write a Python function that given a, b, and c from $ax^2 + bx + c$ returns the solution to this quadratic.

```
def quadratic_equation(a:float, b:float, c:float) -> (float, float)
    """(float, float, float) -> (float, float)
    Returns the solution to ax^2 + bx + c.
    >>> quadratic_equation(1, -5, 6)
    (1.5, 1.0)
    >>> quadratic_equation(10, 0, 1)
    (2.5, 2.5)
    """
    sqrt = lambda x : x**0.5
    disc = sqrt(b**2 - 4*a*c)

    return ( (-b+disc)/4/a, (-b-disc)/4/a )
```