Chapter6_Exercises

May 28, 2017

1 Chapter 6 Exercises

1.1 Exercise 6.1

What does the program print?

```
In [1]: def b(z):
            prod = a(z, z)
            print(z, prod)
            return prod

        def a(x, y):
            x = x + 1
            return x * y

        def c(x, y, z):
            total = x + y + z
            square = b(total)**2
            return square

        x = 1
        y = x + 1
        print(c(x, y+3, x+y))
```

```
9 90
8100
```

1.2 Exercise 6.2

Write a function named ack that evaluates the Ackermann function. Use your function to evaluate ack(3, 4), which should be 125.What happens for larger values of m and n?

```
In [14]: def ack(m,n):
             if not isinstance(m, int) or not isinstance(n,int):
                 print('Undefined for non-integer inputs')
                 return None
             elif m < 0 or n < 0:
```

```
            print('Undefined for negative inputs')
            return None
        elif m == 0:
            return n+1
        elif n == 0:
            return ack(m-1,1)
        else:
            return ack(m-1, ack(m,n-1))

    ack(3,4)
```

```
Out[14]: 125
```

Larger values of m and n correspond to larger outputs and more instances of the function.

1.3 Exercise 6.3

A palindrome is a word that is spelled the same backward and forward, like "noon" and "redivider". Recursively, a word is a palindrome if the first and last letters are the same and the middle is a palindrome.

The following are functions that take a string argument and return the first, last, and middle letters:

```
In [15]: def first(word):
             return word[0]

         def last(word):
             return word[-1]

         def middle(word):
             return word[1:-1]
```

Type these functions into a file named palindrome.py and test them out. What happens if you call middle with a string with two letters? One letter? What about the empty string, which is written '' and contains no letters?

```
In [16]: middle('hi')
```

```
Out[16]: ''
```

```
In [17]: middle('i')
```

```
Out[17]: ''
```

```
In [18]: middle('')
```

```
Out[18]: ''
```

Write a function called is_palindrome that takes a string argument and returns True if it is a palindrome and False otherwise. Remember that you can use the built-in function len to check the length of a string.

```
In [79]: def is_palindrome(s):
             if len(s) <= 0:
                 return True
             if first(s) == last(s):
                 return is_palindrome(middle(s))
             else:
                 return False

         print('The truth is {}'.format(is_palindrome('racecar')))
         print('The truth is {}'.format(is_palindrome('gog')))
         print('The truth is {}'.format(is_palindrome('something something Europear
```

```
The truth is True
The truth is True
The truth is False
```

1.4 Exercise 6.4

A number, a, is a power of b if it is divisible by b and a/b is a power of b. Write a function called is_power that takes parameters a and b and returns True if a is a power of b. Note: you will have to think about the base case

```
In [116]: def is_power(a,b):
              """Returns true if a is a power of b, not-including the zeroth power'
              if a%b == 0 and a <= b:
                  return True
              elif a < b:
                  return False
              else:
                  return is_power(a/b,b)

          print(is_power(16,4))
          print(is_power(20,4))
          print(is_power(256,16))
          print(is_power(30,5))
```

```
True
False
True
False
```

1.5 Exercise 6.5

The greatest common divisor (GCD) of a and b is the largest number that divides both of them with no remainder.

One way to find the GCD of two numbers is based on the observation that if r is the remainder when a is divided by b, then gcd(a, b) = gcd(b,r). As a base case, we can use gcd(a, 0) = a.

Write a function called gcd that takes parameters a and b and returns their greatest common divisor.

```
In [125]: def gcd(a,b):
              if b == 0:
                  return a
              else:
                  return gcd(b, a%b)

          print(gcd(24,36))
          print(gcd(5,5))
          print(gcd(81,72))
```

```
12
5
9
```