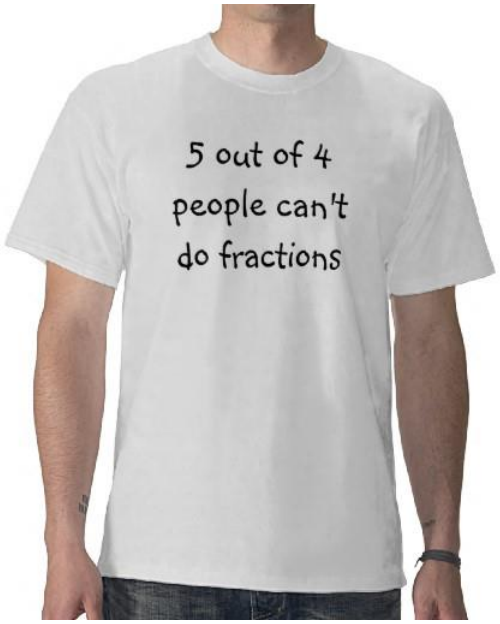
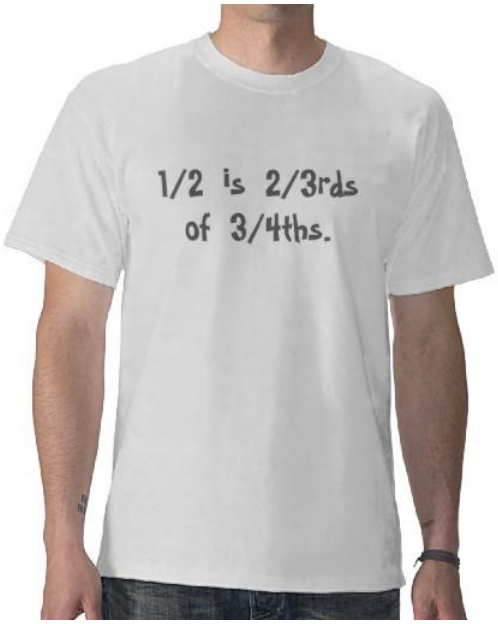
Building Fraction Functions

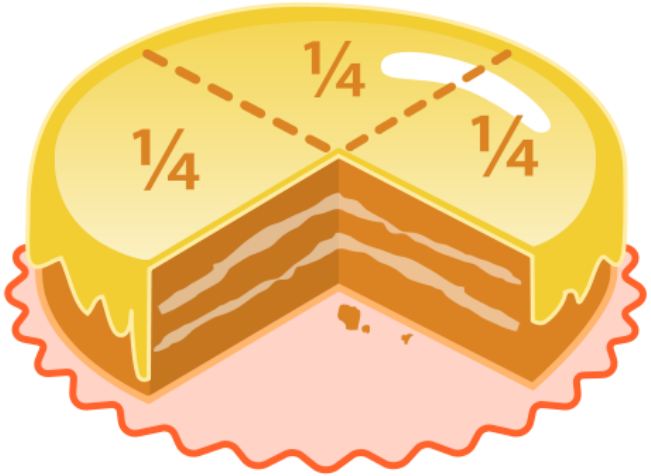




http://www.zazzle.com/fraction+tshirts

Overview

- Functions
  - Practice
    - Fraction Functions



Multiplying fractions

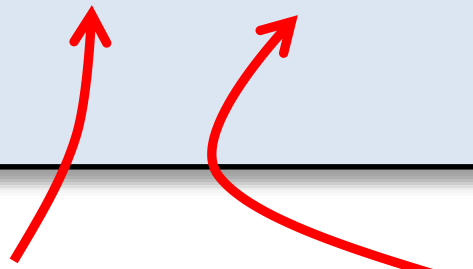
- Goal: Given two fractions, return a new fraction that is the multiplication of the two

```
FractionA = [1, 2]
FractionB = [2, 3]

FractionC = multiply(FractionA, FractionB)

print(str(FractionA) + " * " + str(FractionB) + " = " + str(FractionC))
```

```
% python Fraction.py
[1, 2] * [2, 3] = [1, 3]
```

Multiply Function

```
def multiply(A, B):
    result = [A[0] * B[0], A[1] * B[1]]
    return result
```



Denominator of the first fraction

Denominator of the second fraction

```
FractionC = mutiply(FractionA, FractionB)
```

Multiplying Fractions

- Attempt 1: Close, but not in lowest terms…

```
def multiply(A, B):
    result = [A[0] * B[0], A[1] * B[1]]
    return result

FractionA = [1, 2]
FractionB = [2, 3]

FractionC = multiply(FractionA, FractionB)

print(str(FractionA) + " * " + str(FractionB) + " = " + str(FractionC))
```

```
% python Fraction.py
[1, 2] * [2, 3] = [2, 6]
```

Lowest Terms

- Attempt 2: Add code to reduce to lowest terms

```
def multiply(A, B):
    result = [A[0] * B[0], A[1] * B[1]]
    i = min(abs(result[0]), abs(result[1]))
    if (i == 0):
        return result
    while int(result[0] % i) != 0 or (int(result[1] % i) != 0):
        i -= 1
    result2 = [int(result[0]/i), int(result[1]/i)]
    return result2
```

```
% python Fraction.py
[1, 2] * [2, 3] = [1, 3]
```

Divide Function

- Very similar function for division:

```
def divide(A, B):
    result = [A[0] * B[1], A[1] * B[0]]
    i = min(abs(result[0]), abs(result[1]))
    if (i == 0):
        return result
    while int(result[0] % i) != 0 or (int(result[1] % i) != 0):
        i -= 1
    result2 = [int(result[0]/i), int(result[1]/i)]
    return result2
```

Repeated code is evil™

```
def multiply(A, B):
    result = [A[0] * B[0], A[1] * B[1]]
    i = min(abs(result[0]), abs(result[1]))
    if (i == 0):
        return result
    while int(result[0] % i) != 0 or (int(result[1] % i) != 0):
        i -= 1
    result2 = [int(result[0]/i), int(result[1]/i)]
    return result2
```

Where should this code really live? There are a number of choices, but not here for sure.

We have to repeat it in the `divide()`, `add()`, and `subtract()` functions as well.

Helper Functions

- Add a helper function, `reduce( )`

```
def reduce(A):
    i = min(abs(A[0]), abs(A[1]))
    if (i == 0):
        return A
    while int(A[0] % i) != 0 or (int(A[1] % i) != 0):
        i -= 1
    result2 = [int(A[0]/i), int(A[1]/i)]
    return result2

def divide(A, B):
    result = [A[0] * B[1], A[1] * B[0]]
    result2 = reduce(result)
    return result2
```

Fill in the Missing Code

```
def multiply(A, B):
    result = (A[0] * B[0], A[1] * B[1])
    result2 = reduce(result)
    return result2

def equals(A, B):



def reciprocal(A):



def add(A, B):



def subtract(A, B):

```

Summary

- Functions
  - Practice
    - Fraction Functions