from operator import add, sub

def a\_plus\_abs\_b(a, b):

    """Return a+abs(b), but without calling abs.

    >>> a\_plus\_abs\_b(2, 3)

    5

    >>> a\_plus\_abs\_b(2, -3)

    5

    >>> # a check that you didn't change the return statement!

    >>> import inspect, re

    >>> re.findall(r'^\s\*(return .\*)', inspect.getsource(a\_plus\_abs\_b), re.M)

    ['return f(a, b)']

    """

    if b < 0:

        f = sub

    else:

        f = add

    return f(a, b)

def two\_of\_three(x, y, z):

    """Return a\*a + b\*b, where a and b are the two smallest members of the

    positive numbers x, y, and z.

    >>> two\_of\_three(1, 2, 3)

    5

    >>> two\_of\_three(5, 3, 1)

    10

    >>> two\_of\_three(10, 2, 8)

    68

    >>> two\_of\_three(5, 5, 5)

    50

    >>> # check that your code consists of nothing but an expression (this docstring)

    >>> # a return statement

    >>> import inspect, ast

    >>> [type(x).\_\_name\_\_ for x in ast.parse(inspect.getsource(two\_of\_three)).body[0].body]

    ['Expr', 'Return']

    """

    return min(x,y,z)\*min(x,y,z) + max(min(x,y),min(y,z),min(x,z))\*max(min(x,y),min(y,z),min(x,z))

def largest\_factor(n):

    """Return the largest factor of n that is smaller than n.

    >>> largest\_factor(15) # factors are 1, 3, 5

    5

    >>> largest\_factor(80) # factors are 1, 2, 4, 5, 8, 10, 16, 20, 40

    40

    >>> largest\_factor(13) # factor is 1 since 13 is prime

    1

    """

    "\*\*\* YOUR CODE HERE \*\*\*"

    j=0

    for i in range(1,n):

        if (n%i==0):

            j=i

    return j

def if\_function(condition, true\_result, false\_result):

    """Return true\_result if condition is a true value, and

    false\_result otherwise.

    >>> if\_function(True, 2, 3)

    2

    >>> if\_function(False, 2, 3)

    3

    >>> if\_function(3==2, 3+2, 3-2)

    1

    >>> if\_function(3>2, 3+2, 3-2)

    5

    """

    if condition:

        return true\_result

    else:

        return false\_result

def with\_if\_statement():

    """

    >>> result = with\_if\_statement()

    47

    >>> print(result)

    None

    """

    if cond():

        return true\_func()

    else:

        return false\_func()

def with\_if\_function():

    """

    >>> result = with\_if\_function()

    42

    47

    >>> print(result)

    None

    """

    return if\_function(cond(), true\_func(), false\_func())

def cond():

    "\*\*\* YOUR CODE HERE \*\*\*"

    return False

def true\_func():

    "\*\*\* YOUR CODE HERE \*\*\*"

    print(42)

def false\_func():

    "\*\*\* YOUR CODE HERE \*\*\*"

    print(47)

def hailstone(n):

    """Print the hailstone sequence starting at n and return its

    length.

    >>> a = hailstone(10)

    10

    5

    16

    8

    4

    2

    1

    >>> a

    7

    """

    "\*\*\* YOUR CODE HERE \*\*\*"

    i=0

    while (n>1):

        print(n)

        if (n%2==0):

            n=n//2

            i=i+1

        else:

            n=n\*3+1

            i=i+1

    if (n==1):

        print(n)

        i=i+1

    return i