

CS101. A6 - Triangle

Given lengths of 3 sides of a Triangle, identify its type amongst the following.

- ★ For a shape to be a triangle at all, all sides have to be of length > 0 , and the sum of the lengths of any two sides must be greater than or equal to the length of the third side.
- ★ An equilateral triangle has all three sides the same length.
- ★ Isosceles triangle if two sides of the triangle are the same. An isosceles triangle has at least two sides the same length.
- ★ A scalene triangle has all sides of different lengths.
- ★ Degenerate triangle: sum of the lengths of two sides equals that of the third. It has zero area and looks like a single line

Write code inside 5 functions (each for a type of a triangle) to ascertain the type. Examples:

- `is_equilateral(2, 2, 2)` returns True, `is_equilateral(2, 3, 4)` returns False
- `is_isosceles(3, 4, 4)` will return True. `is_isosceles(4, 4, 4)` will return True. `isosceles(2, 3, 4)` will return False
- `is_scalene(5, 4, 6)` returns True, `is_scalene(7, 3, 2)` returns False
- `is_degenerate(2,4,6)` returns True, `is_degenerate(2,4,3)` returns False
- More in the Test Cases Table.

Code Template (grains.py)

```
"""
Numbers - Triangle example
"""
def is_triangle(side1, side2, side3):
    pass

def is_equilateral(side1, side2, side3):
    pass

def is_isosceles(side1, side2, side3):
    pass

def is_scalene(side1, side2, side3):
    pass

def is_degenerate(side1, side2, side3):
    pass
```

Test Cases

Function	Inputs	Output	Remarks
<code>is_triangle</code>	0, 0, 0	False	
<code>is_triangle</code>	10, 0, 0	False	
<code>is_triangle</code>	10, 20, 0	False	
<code>is_triangle</code>	10, 20, 30	True	
<code>is_triangle</code>	10, 1, 1	False	
<code>is_triangle</code>	10, 5, 5	True	
<code>is_triangle</code>	10, 5, 4	False	

is_equilateral	10, 10, 10	True	
is_equilateral	0, 0, 0	False	
is_equilateral	5, 5, 5	True	
is_equilateral	5, 5, 4	False	
is_equilateral	5, 3, 4	False	
is_equilateral	0.5, 0.5, 0.5	True	
is_equilateral	10, 1, 1	False	
is_isosceles	3, 4, 4	True	
is_isosceles	4, 3, 4	True	
is_isosceles	4, 4, 3	True	
is_isosceles	4, 4, 4	True	
is_isosceles	3, 4, 5	False	
is_isosceles	0, 0, 0	False	
is_isosceles	10, 1, 1	False	
is_isosceles	1, 10, 1	False	
is_isosceles	1, 1, 10	False	
is_isosceles	0.5, 0.5, 0.4	True	
is_scalene	5, 4, 6	True	
is_scalene	4, 4, 4	False	
is_scalene	4, 4, 3	False	
is_scalene	10, 1, 1	False	
is_scalene	0.5, 0.4, 0.6	True	
is_degenerate	0, 0, 0	False	
is_degenerate	10, 5, 5	True	
is_degenerate	10, 15, -5	raise ValueError("Side cannot be negative.")	
is_triangle	-10, 20, 30		
is_isosceles	-10, 20, 30		
is_scalene	-10, 20, 30		
is_equilateral	-10, 20, 30		

CS101. A7 - Isogram

Determine if a word or phrase is an isogram.

An isogram (also known as a "non-pattern word") is a word or phrase without a repeating letter, however spaces and hyphens are allowed to appear multiple times. Examples of isograms: lumberjacks, background, downstream, six-year-old, isogram.

The word isograms, however, is not an isogram, because the s repeats.

Input: A word or a phrase (String)

Output – True / False

True (if the supplied input is an Isogram)

False (if the supplied input is NOT an Isogram)

Examples:

- "" (Empty String)
- uncopyrightable (longest isogram)
- First# Clan! (string contains punctuation marks)
- "BackGround" (String contains upper- and lowercase letter)
- Non-isograms containing single letter repetition to several letters repeating.

Code Template (isogram.py)

```
"""
Isogram
"""

def is_isogram(string):
    pass
```

Sample output

```
>>> print(is_isogram(""))
False
>>> print(square(4))
8
```

Test Cases

Function	Inputs	Output	Remarks
isogram	""	True	Empty String
isogram	"isogram"	True	
isogram	"eleven"	False	
isogram	"zzyzx"	False	
isogram	"subdermatoglyphic"	True	
isogram	"Alphabet"	False	

isogram	“alphAbet”	False	
isogram	“thumbscrew-japingly”	True	
isogram	“Dictionary”	False	
isogram	“thumbscrew-jappingly”	False	
isogram	“six-year-old”	True	
isogram	“Emily Jung Schwartzkopf”	True	
isogram	“Accenture”	False	
isogram	“BackGround”	True	
isogram	“First# Clan!”	True	
isogram	“uncopyrightable”	True	

CS101. A8 - Sum of digits

Write a function to return the sum of the digits of an integer. Raise a ValueError if any other datatype is input. Ignore the sign.

Examples

- `sum_of_digits(10)`, `sum_of_digits(10000000)`, `sum_of_digits(010)` will all return 1
- `sum_of_digits(10.0)` will raise `ValueError "Invalid input type. Expecting an integer."`

Tasks in the Assignment

- Implement one function: `sum_of_digits(number)`
- Input: an Integer.
- Output: Sum of the digits.

Code Template (sum_of_digits.py)

```
'''
Sum of digits
'''

def sum_of_digits(number):
    pass
```

Test Cases

Function	Inputs	Output	Remarks
sum_of_digits	-1.0	ValueError	
sum_of_digits	10	1	
sum_of_digits	34	7	
sum_of_digits	100	1	
sum_of_digits	10000	1	

Ref: https://en.wikipedia.org/wiki/Fizz_buzz

Your task is to convert a number into a string that contains raindrop sounds corresponding to certain potential factors. A factor is a number that evenly divides into another number, leaving no remainder. The simplest way to test if a one number is a factor of another is to use the modulo operation.

The rules of raindrops are that if a given number:

- has 2 as a factor, add 'rim jhim' to the result.
- has 3 as a factor, add 'jal tarang' to the result.
- has 5 as a factor, add 'baadal' to the result.
- has 7 as a factor, add 'chalte hain' to the result.
- *does not* have any of 2, 3, 5, or 7 as a factor, the result should be the digits of the number.

Examples

- 7 has 7 as a factor and not 2, 3 or 5. Result: "chalte hain"
- 34 has 2 as a factor and is not factored by 3, 5, or 7, so the result would be "rim jhim".
- 15 has 3 and 5 as factors and not 2 or 7. Result: "jal tarang baadal".
- 28 has 2 and 7 as factors, but not 3 or 5, so the result would be "rim jhim chalte hain".
- 30 has 2, 3 and 5 as factors, but not 7, so the result would be "rim jhim jal tarang chalte hain".
- 13 no factors in 2, 3, 5 or 7. Result "13".
- Input is valid if it is a positive number ≥ 2 . Any other number given as input: `raise ValueError` "Invalid input."

Tasks in the Assignment

- Implement one function: `convert(number)`
- Input: Integer
- Output: String.

Code Template (raindrops.py)

```
'''
Raindrops
'''

def convert(number):
    pass
```

Sample output

```
print(convert(34))
rim jhim

print(convert(210))
rim jhim jal tarang baadal chalte hain
```

Test Cases

Function	Inputs	Output	Remarks
----------	--------	--------	---------

convert	-1	ValueError	
convert	7	chalte hain	
convert	13	13	
convert	15	jal tarang baadal	
convert	28	rim jhim chalte hain	
convert	30	rim jhim jal tarang baadal	
convert	34	rim jhim	
convert	210	rim jhim jal tarang baadal chalte hain	
