

# Table of content

progression .....	2
collatz .....	3
geometric_seq .....	4
permutations .....	5

# progression

## Your mission\_EN

Write a function that takes an integer  $n$  as parameter, and returns the sum of  $1/2 + 2/3 + 3/4 + \dots + n/n+1$ , rounded up to two decimals.

If the argument passed to the function is not an integer, the function will return "Invalid Value".

Your function must be named `progression` and in the file named `progression.py` (and in a directory named `progression`).

## Your mission\_VN

Viết một hàm nhận một số nguyên làm tham số, và trả về tổng của:  $1/2 + 2/3 + 3/4 + \dots + n/n+1$ , làm tròn đến 2 số thập phân.

Nếu tham số truyền vào không phải là số nguyên, hàm sẽ trả về "Invalid Value".

Hàm phải được đặt tên là `progression` trong tập tin đặt tên là `progression.py` (và trong thư mục đặt tên là `progression`).

## Example

```
1     def progression(n):
2         """ Takes an integer as argument and returns the following
progression: 1/2 + 2/3 + 3/4 + ... + n/n+1
3         >>> progression(1)
4         0.5
5         >>> progression(12)
6         9.82
7         >>> progression("nope")
8         'Invalid Value'
9         """
```

# collatz

## Your mission\_EN

Write a function that takes a positive integer  $n$  as parameter, and returns the number of steps needed to reach 1 using the following process: if  $n$  is even, divide it by 2; if  $n$  is odd, multiply it by 3 and add 1. This is called the Collatz conjecture (or Syracuse problem).

If the argument passed to the function is not a strictly positive integer, the function will return -1.

Your function must be named `collatz` and in the file named `collatz.py` (and in a directory named `collatz`).

## Your mission\_VN

Viết một hàm nhận số nguyên dương  $n$  làm tham số, và trả về số bước cần thiết để đạt được 1 sử dụng quy trình như sau: Nếu  $n$  là số chẵn, chia  $n$  cho 2; Nếu  $n$  là số lẻ, nhân  $n$  cho 3 và cộng thêm 1. Quy trình này được gọi là ước lượng Collatz (hay bài toán Syracuse).

Nếu tham số truyền vào hàm không phải là số nguyên dương, hàm sẽ trả về -1.

Hàm phải được đặt tên là `collatz` trong tập tin đặt tên là `collatz.py` (và trong thư mục đặt tên là `collatz`)

## Example

```
1     def collatz(n):
2         """ Takes a positive integer as argument and returns the
3             number of steps needed to reach 1 using the following process: if n is
4             even, divide it by 2; if n is odd, multiply it by 3 and add 1.
5
6             >>> collatz(1)
7             0
8             >>> collatz(9)
9             19
10            >>> collatz(21)
11            7
12            >>> collatz(0)
13            -1
14            """
```

# geometric\_seq

## Your mission\_EN

Write a function that takes a list of integers as argument, and returns whether the sequence of integers constitutes a geometric sequence.

By definition, a geometric sequence is a sequence of numbers where each term after the first is found by multiplying the previous one by the same fixed, non-zero number.

If the argument passed to the function is not a list of integers, the function will return "Invalid Value".

Your function must be named `geometric_seq` and in the file named `geometric_seq.py` (and in a directory named `geometric_seq`).

## Your mission\_VN

Viết một hàm mà nhận một danh sách các số nguyên làm tham số, và trả kết quả chuỗi nguyên có tạo thành một cấp số nhân hay không.

Theo định nghĩa, cấp số nhân là chuỗi các số mà ở đó mỗi số sau số hạng đầu tiên bằng tích của số hạng trước nó với một số không đổi, khác 0.

Nếu tham số truyền vào hàm không phải là danh sách các số nguyên, hàm sẽ trả về "Invalid Value".

Hàm phải được đặt tên là `geometric_seq` trong tập tin đặt tên là `geometric_seq.py` (và trong thư mục đặt tên là `geometric_seq`).

## Example

```
1 def geometric_seq(lst):
2     """ Takes a list of integers as argument and returns whether
the sequence is geometric.
3     >>> geometric_seq([3, 6, 12, 24])
4     True
5     >>> geometric_seq([1, -3, 9, -27])
6     True
7     >>> geometric_seq([0, 2, 4, 8])
8     False
9     >>> geometric_seq([2, 4, "8"])
10    'Invalid Value'
11    """
```

# permutations

## Your mission\_EN

Write a function that takes a list of characters as parameter, and returns all their possible permutations as a list of strings. Note that in a permutation, the order of the elements matter.

If the argument passed to the function is not a list of characters, the function will return "Invalid Value".

Your function must be named permutations and in the file named permutations.py (and in a directory named permutations).

## Your mission\_VN

Viết một hàm nhận một danh sách các ký tự làm tham số, và trả về tất cả những hoán vị có thể có của chúng như là danh sách của các chuỗi. Lưu ý rằng trong một hoán vị, thì thứ tự của các phần tử là quan trọng.

Nếu tham số truyền vào hàm không phải là danh sách các ký tự, hàm sẽ trả về "Invalid Value".

Hàm phải được đặt tên là permutations trong tập tin đặt tên là permutations.py (và trong thư mục đặt tên là permutations).

## Example

```
1     def permutations(lst):
2         """ Takes a list of characters as argument and returns all
their possible permutations as a list of strings
3         >>> permutations(["a", "i", "o"])
4         ['aio', 'aoi', 'iao', 'ioa', 'oai', 'oia']
5         >>> permutations(["eho", "l", "h"])
6         'Invalid Value'
7         """
```