

Exercise 1: Euclid's Algorithm (Total: 12 points)

Euclid's algorithm finds the greatest common divisor (gcd) of two integers a and b . The procedure is as follows: beginning with $r_{-1} = a$ and $r_0 = b$, choose successive values

$$r_{k+1} = r_{k-1} \bmod r_k; k = 0, \dots$$

until for some $k = N$ you find $r_N = 0$. The gcd of a and b is then r_{N-1} .

Write a Python function to compute the gcd of two numbers. Ensure that you begin the iteration with integers (i.e. by casting floats to integer).

Test your function on the pairs:

- `gcd(35, 42) = 7`
- `gcd(1071, 462) = 21`
- `gcd(2232, 5952) = 744`
- `gcd(6831, 15384) = 3`

```
In [3]: #Code Here
def gcd(a,b):
    a,b = int(a), int(b)

    while b != 0:
        a,b = b,a %b
    return a

test_cases = [
    (35, 42),
    (1071, 462),
    (2232, 5952),
    (6831, 15384)
]

results = {f"gcd({a}, {b})": gcd(a, b) for a, b in test_cases}
results
```

```
Out[3]: {'gcd(35, 42)': 7,
         'gcd(1071, 462)': 21,
         'gcd(2232, 5952)': 744,
         'gcd(6831, 15384)': 3}
```

Exercise 2: Optional Arguments and Multiple Return Values (Total: 8 points)

Write a Python function that will take a list of numbers and return various statistics. The function should take one required argument (the list of numbers) and one optional argument (a tuple or list of strings naming the statistics to compute). The default value for the optional argument should be `("mean",)`. The allowed strings are (assuming the N elements of the supplied list are L_i):

- `"mean"` return the mean value of the list
- `"sum"` return the sum of the list values
- `"rms"` return the root-mean-square, i.e. $\sqrt{1/N * \sum(L_i^2 \text{ for } i \text{ in range}(N))}$
- `"max"` return the maximum
- `"min"` return the minimum

The return value should be a tuple containing the requested statistics in the same order as they appear in the request tuple.

Test your function on the list:

```
[-34.2, 17.1, 3.8, 1.4, 22.6, 19.4, 15.9]
```

with different combinations of requested statistics (or no request).

```
In [9]: #Code Here
from math import sqrt

def compute(numbers, stats=("mean",)):
    print(stats, "VALUE IN STATS")
    res = []
    N = len(numbers)
    if (len(stats) == 0) :
        mean_val = sum(numbers) # N THIS IS NOT NEEDED SINCE WE WILL TEST DIFFERENT
        res.append(mean_val) #but it works if we pass an empty string
        # BUT IF THE USER STILL PASSES IN A BLANK STRING WE HAVE THIS SAFETY CHECK
    for i in stats:
        if (i == "mean" ):
            mean_val = sum(numbers) / N
            res.append(mean_val)
        elif(i == "sum"):
            sum_val = sum(numbers)
            res.append(sum_val)
        elif(i == "rms"):
            rms_val = sqrt(sum(x**2 for x in numbers)/ N)
            res.append(rms_val)
        elif(i == "max"):
            max_val = max(numbers)
            res.append(max_val)
        elif(i == "min"):
            min_val = min(numbers)
            res.append(min_val)
    return tuple(res)
test_list = [-34.2, 17.1, 3.8, 1.4, 22.6, 19.4, 15.9]
test_cases = [
    (),
```

```

        ("mean", "sum"),
        ("rms", "max", "min"),
        ("max", "min", "mean", "sum", "rms")
    ]
    results = {str(case): compute(test_list, case) if len(case) != 0 else compute(test_
    results

```

```

('mean',) VALUE IN STATS
('mean', 'sum') VALUE IN STATS
('rms', 'max', 'min') VALUE IN STATS
('max', 'min', 'mean', 'sum', 'rms') VALUE IN STATS

```

```

Out[9]: {'()': (6.571428571428571,),
        "('mean', 'sum')": (6.571428571428571, 46.0),
        "('rms', 'max', 'min')": (19.340483078617392, 22.6, -34.2),
        "('max', 'min', 'mean', 'sum', 'rms')": (22.6,
        -34.2,
        6.571428571428571,
        46.0,
        19.340483078617392)}

```

```

In [ ]: %%capture
# Here we use a script to generate pdf and save it to google drive.

# After executing this cell, you will be asked to link to your GoogleDrive account.
# Then, the pdf will be generated and saved to your GoogleDrive account and you nee

from google.colab import drive
drive.mount('/content/drive')
# install tex; first run may take several minutes
! apt-get install texlive-xetex
# file path and save location below are default; please change if they do not match
! jupyter nbconvert --output-dir='/content/drive/MyDrive/' '/content/drive/MyDrive/

```