Assignment 4 Analysis

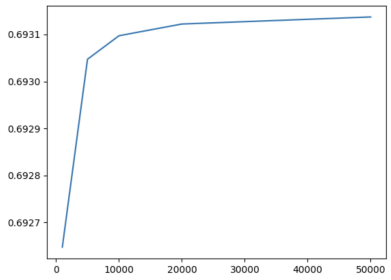
Manciu Patricia-Ariana → 914

```
import math
from matplotlib import pyplot as plt


def calculate_sum(n: int):
    summ = 0
    for k in range(1, n + 1):
        summ += ((-1) ** (k + 1)) / k
    return summ


ln2 = math.log(2)


values = [1000, 5000, 10000, 20000, 50000]

print(f"The value of ln2 = {ln2}")
sum_values = []
for n in values:
    s = calculate_sum(n)
    sum_values.append(s)
    print(f"For n = {n}, the sum ∑n≥1 (−1)^n+1 / n = {s}")

fig, ax = plt.subplots()
ax.plot(values, sum_values)
plt.show()


def calculate_rearranged(p: int, q: int):
    rearranged_sum = 0
    for k in range(1, p + 1):
        rearranged_sum += 1 / (2 * k - 1)
        # the odd numbers are positive
    for k in range(1, q + 1):
        rearranged_sum -= 1 / (2 * k)
        # the even numbers are negative
    return rearranged_sum


p_val = [500, 2500, 5000, 10000, 25000]
q_val = [500, 2500, 5000, 10000, 25000]

for p in p_val:
```

```
    result_p = []
    for q in q_val:
        r_s = calculate_rearranged(p, q)
        result_p.append(r_s)
        print(f"For p = {p}, q = {q} the rearranged sum is {r_s}")
    fig2, ax2 = plt.subplots()
    ax2.plot(q_val, result_p)
    plt.show()
```

The outcome for the sum computed in it’s original order:

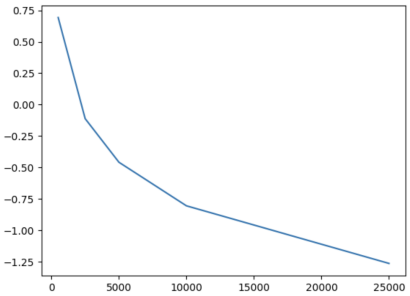
```
The value of ln2 = 0.6931471805599453
For n = 1000, the sum ∑n≥1 (-1)^n+1 / n = 0.6926474305598223
For n = 5000, the sum ∑n≥1 (-1)^n+1 / n = 0.6930471905599515
For n = 10000, the sum ∑n≥1 (-1)^n+1 / n = 0.6930971830599583
For n = 20000, the sum ∑n≥1 (-1)^n+1 / n = 0.6931221811849583
For n = 50000, the sum ∑n≥1 (-1)^n+1 / n = 0.6931371806599672
```



The outcome for the sum computed depending on p and q:

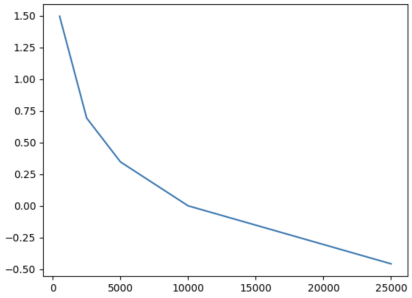
p = 500

```
For p = 500, q = 500 the rearranged sum is 0.6926474305598131
For p = 500, q = 2500 the rearranged sum is -0.11167168565717096
For p = 500, q = 5000 the rearranged sum is -0.4581952809371454
For p = 500, q = 10000 the rearranged sum is -0.8047438724671223
For p = 500, q = 25000 the rearranged sum is -1.262874238754193
```



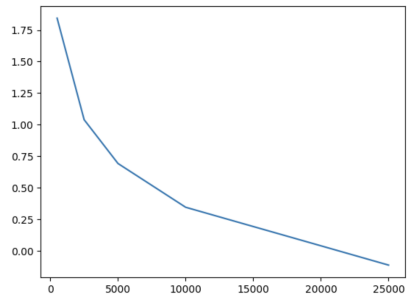
p = 2500

```
For p = 2500, q = 500 the rearranged sum is 1.497366306776921
For p = 2500, q = 2500 the rearranged sum is 0.6930471905599377
For p = 2500, q = 5000 the rearranged sum is 0.3465235952799645
For p = 2500, q = 10000 the rearranged sum is -2.4996250008657917e-05
For p = 2500, q = 25000 the rearranged sum is -0.45815536253708977
```



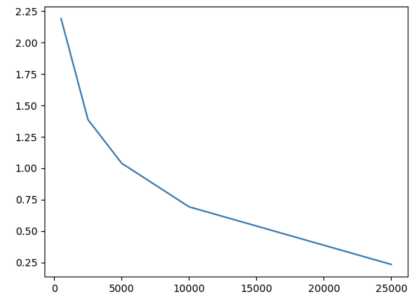
p = 5000

```
For p = 5000, q = 500 the rearranged sum is 1.8439398945568968
For p = 5000, q = 2500 the rearranged sum is 1.0396207783399165
For p = 5000, q = 5000 the rearranged sum is 0.6930971830599475
For p = 5000, q = 10000 the rearranged sum is 0.34654859152997325
For p = 5000, q = 25000 the rearranged sum is -0.1115817747571041
```



p = 10000

```
For p = 10000, q = 500 the rearranged sum is 2.190513484211851
For p = 10000, q = 2500 the rearranged sum is 1.3861943679948663
For p = 10000, q = 5000 the rearranged sum is 1.0396707727148835
For p = 10000, q = 10000 the rearranged sum is 0.6931221811849057
For p = 10000, q = 25000 the rearranged sum is 0.23499181489782153
```



p = 25000

```
For p = 25000, q = 500 the rearranged sum is 2.6486588499739403
For p = 25000, q = 2500 the rearranged sum is 1.8443397337569538
For p = 25000, q = 5000 the rearranged sum is 1.497816138476971
For p = 25000, q = 10000 the rearranged sum is 1.1512675469470002
For p = 25000, q = 25000 the rearranged sum is 0.693137180659934
```