# Problem 5

First, we sum the probabilities

$$p_n = \frac{0.1}{n}$$

for $n \in [0, 12367] \subset \mathbb{Z}^+$ to ensure the result is normal (*close to 1*)

In [1]:
```
probSumNow = 0
for i in range(12367):
    probSumNow += 0.1 / (i + 1)

print(probSumNow)
```

```
1.0000043008275796
```

Since the sum of the probabilities is reasonably close to 1, we proceed to calculate the entropies using

$$H(x) = -\sum_{n=1}^{12367} \left\{ \frac{0.1}{n} \log_2 \left[ \frac{0.1}{n} \right] \right\}$$

to obtain

In [2]:
```
import math

entropySumNow = 0
for i in range(12367):
    entropySumNow += - (0.1 / (i + 1)) * math.log2(0.1 / (i + 1))

print(entropySumNow)
```

```
9.71625847652071
```

Based on this result and the results in **Problem 1** (***parts A and B***), the entropy of words is bounded above by the entropy of Bi-Grams and below by the entropy of single characters. This would seem to imply that using at most two letters for determining frequencies for Huffman encoding is probably ideal.