A1

In [1]:
```
# Standard imports
import numpy as np
np.seterr(all='ignore');  # allows floating-point exceptions
import matplotlib.pyplot as plt
```

Q3: FPNS $\mathcal{F}(6, 6, -6, 6)$

(a)

The largest value in $\mathcal{F}$ is:

0.555555×6^6

(b)

The value in our number system is:

0.545335×6^{-2}

(c)

From class we learned that we can express the machine epsilon as:

E = $\frac{1}{2} B^{1-t}$

Plugging in our values give us:

E = $\frac{1}{2} \times 6^{1-6}$

E = 0.500000×6^5

(d)

The total number of normalized numbers that can be expressed is:

= 2 (sign) $\times 5$ (first digit) $\times 6^5$ (next 5 digits) $\times 13$ (exponent)

= 1010880

Therefore we need to calculate the amount of normalized numbers that are less then:

= 0.100000 \times 6^1

will be greater then 1. The number of combinations is:

= 2 (sign) $\times 5$ (first digit) $\times 6^5$ (next 5 digits) $\times 7$ (exponent)

= 544320

Which gives us a fraction of:

= $\frac{544320}{1010880}$

= $\frac{7}{13}$

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```