ASCII Art Compression

Use the "encodeString" and "decodeString" functions from the Chapter 4 challenge, provided below

Read in the ASCII art text file 10_04_challenge_art.txt and write it back to a new file that has a smaller file size than the original file. For example, the original 10_04_challenge_art.txt has a file size of 2.757kB (or 2,757 ASCII characters).

- Any compression is great!
- Is there any way you could get this file to 1kb?
- Less than 1kb?

After compressing the file, make sure to check your work by opening and decoding it again!

In [3]:
```
import os

import json

# Encodes as a list of (char, count) tuples
def encodeString(stringVal):
    encodedList = []
    prevChar = None
    count = 0
    for char in stringVal:
        if prevChar != char and prevChar is not None:
            encodedList.append((prevChar, count))
            count = 0
        prevChar = char
        count = count + 1
    encodedList.append((prevChar, count))
    return encodedList

def decodeString(encodedList):
    decodedStr = ''
    for item in encodedList:
        try:
            decodedStr = decodedStr + item[0] * item[1]
        except:
            print(item)
    return decodedStr
```

In [4]:

```
def encodeFile(filename, newFilename):
    with open(filename) as f:
        data = encodeString(f.read())

    with open(newFilename, 'w') as f:
        f.write(json.dumps(data))


def decodeFile(filename):
    with open(filename) as f:
        data = f.read()
    return decodeString(json.loads(data))



print(f'Original file size: {os.path.getsize("10_04_challenge_art.txt")}')
encodeFile('10_04_challenge_art.txt', '10_04_challenge_art_encoded.txt')
print(f'New file size: {os.path.getsize("10_04_challenge_art_encoded.txt")}')
print(decodeFile('10_04_challenge_art_encoded.txt'))
```

```
Original file size: 2757
New file size: 2441
```

Better Solution

In [5]:

```
# [('A', 1), ('B', 80), ('C', 10)]
# becomes A|1~B|80~C|10
def encodeFile(filename, newFilename):
    with open(filename) as f:
        data = encodeString(f.read())

    data = [f'{char}|{count}' for char, count in data]

    with open(newFilename, 'w') as f:
        f.write('~'.join(data))


def decodeFile(filename):
    with open(filename) as f:
        data = f.read()

    pairs = data.split('~')
    pairs = [p.split('|') for p in pairs]
    pairs = [(p[0], int(p[1])) for p in pairs]
    return decodeString(pairs)
```

In [6]:
```
print(f'Original file size: {os.path.getsize("10_04_challenge_art.txt")}')
encodeFile('10_04_challenge_art.txt', '10_04_challenge_art_encoded.txt')
print(f'New file size: {os.path.getsize("10_04_challenge_art_encoded.txt")}')
print(decodeFile('10_04_challenge_art_encoded.txt'))
```

```
Original file size: 2757
New file size: 1007
```

Better-er Solution

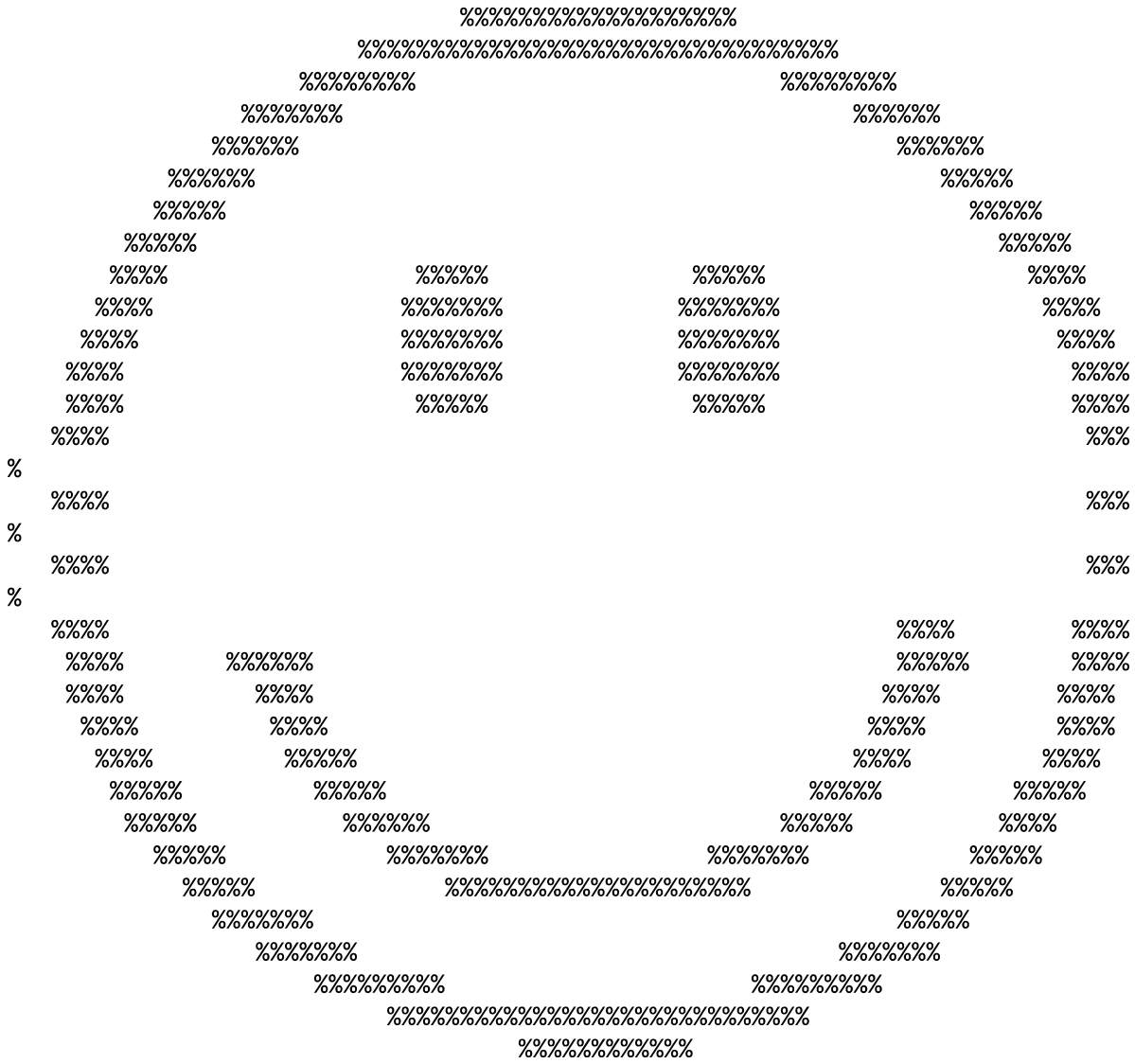
In [7]:
```
def encodeFile(filename, newFilename):
    with open(filename) as f:
        data = encodeString(f.read())
    output = bytearray()
    for item in data:
        # Character byte
        output.extend(bytes(item[0], 'utf-8'))
        # Integer count byte
        output.extend(item[1].to_bytes(1, 'big'))
    with open(newFilename, 'wb') as binary_file:
        # Write bytes to file
        binary_file.write(output)


def decodeFile(filename):
    with open(filename, 'rb') as f:
        data = f.read()
        # Split data into pairs
        bytePairs = [data[i:i+2] for i in range(0, len(data), 2)]
        encodedList = []
        for bytePair in bytePairs:
            encodedList.append((bytePair[:1].decode('utf-8'), int.from_bytes(by
        return decodeString(encodedList)
```

In [8]:
```
print(f'Original file size: {os.path.getsize("10_04_challenge_art.txt")}')
encodeFile('10_04_challenge_art.txt', '10_04_challenge_art_encoded.aa')
```

```
Original file size: 2757
```

In [9]:
```
print(f'New file size: {os.path.getsize("10_04_challenge_art_encoded.aa")}')
print(decodeFile('10_04_challenge_art_encoded.aa'))
```

```
New file size: 466
```



In [ ]:
```

```