Practical 9 Compression

RunLengthEncoding for a given String is implemented in package Practical 9-DataCompression - it takes a string like aaabbrrrrraaaaccccc
And return 3a2b5r4a5c

Binary Compression:

Part 1: Total bits in the binary file 4runs.bin

4runs.bin produces the following output:

```
0000000000000001111111000000011111111111
40 bits
```

Part 2: Compressing the file with runLength

```
0000111100000111
0000011100001011
32 bits
```

Compression ratio : 32/40 = 80% compression ratio

Part 3: Comparing file compression

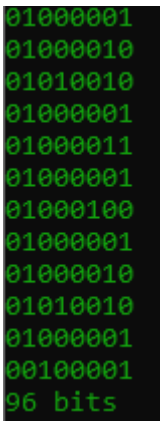
4runsrle.bin produces the following output

```
00001111000001110000011100001011
32 bits
```

Meaning that the compression ratio has stayed the same at 32/40 = 80% compression ratio.

Ascii Compression:

Part1: Total bits in abra.txt



We can see we output 96 bits.

Part2: Run Length Encoding

We get an extremely long Output of 416 bits
Compression ratio = 416/96 = 433.33 %

Part 3 Explanation.
RunLength encoding only works when we have long runs of bits of the same value (0 or 1) this works in the case of our .bin files. However we see in the case of ascii there are no long runs and this means RunLength will use more bits to compress a character.

E.g ascii A = 01000001

RunLength of 010001 =
0000001 (one 0)
0000001 (one 1)
0000101(five 0s)
0000001 (one 1)

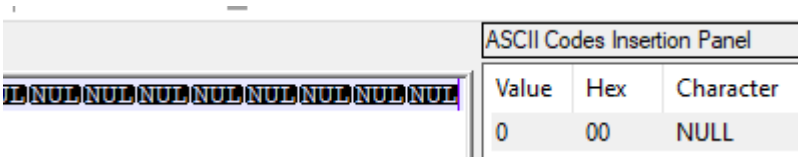
This turns 8 bits into 32 bits

Part 3: Creating text file suitable for ascii

ASCII	Decimal	Hexadecimal	Octal	Binary
null	0	0	0	0

255	FF	377		11111111

We can see both the null character and Delete(255) are both the longest runs and would lend themselves best to runLength Encoding.



NotePad++ allows insertion of nuls into a text file which means there will be a continuous run of 0s

This OptimalAscii.txt file has 344 bits all of which are 0

```
00000000
344 bits
```

When runLengthEncoded we get

```
11111111
00000000
01011001
24 bits
```

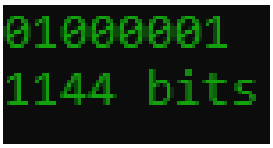
Which results in a compression ratio of 24/344 of 6.98%

Bitmap Compression:

Part 1: bits in file q32x48.bin

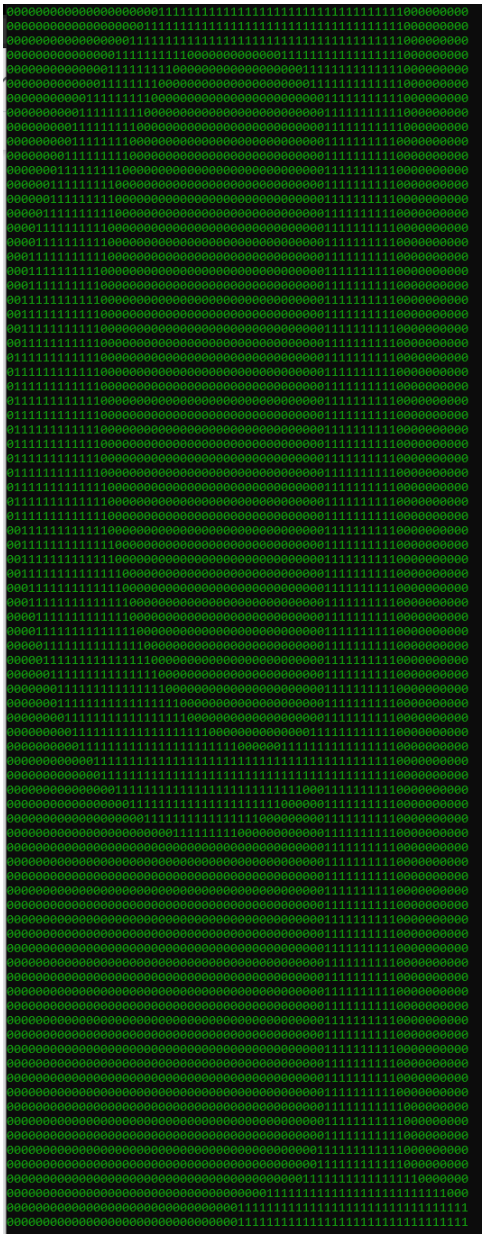
We can see it outputs 1536 bits many of which are long runs of 0s or 1s

Part2 : Compressing the file with RLE



Part3: Compression ratio
Compression ratio = 1144/1536 = 74.48%

Part 4 Compression: 6144 bits



Part 2: compression : 2296 bits

Part 3 Compression ratio

Compression ratio = 2296/6144 = 37.37%

We can see clearly that the difference between image 1 and image 2 is their height and width, this means in image 2 there are longer runs of 1s and 0s allowing for greater compression.