Kier McGuirk - 18752609

Lab 9

Binary Compression

1. java **BinaryDump** 40 < 4runs.bin

   Num bits: 40 bits

2. java RunLength **-** < 4runs.bin **|** java BinaryDump
   Num bits: 32

   Compression ratio = 32 / 40 = 0.8

3. java **BinaryDump** 40 < 4runsrle.bin
   num bits: 32

ASCII Compression

1. java BinaryDump 8 < abra.txt
   96 bits

2. java RunLength **-** < abra.txt **|** java BinaryDump 8
   416

   Compression ratio = 418 /96 = 4.4

   I believe this occurs because RunLength compression isn't suitable for ASCII compression. It is more suitable for bitmap graphics as its data is stored as a series of 1s and 0s.

3. I created my own text file called "random.txt" – containing a random sequence of chars
   - Java BinaryDump 8 < random.txt
     - 808 bits
   - Java RunLength -< random.txt | java BinaryDump 8
     - 3656 bits
   - Compression Ratio = 3656 / 808 = 4.5

Bitmap Compression

Step 1: Use BinaryDump to find out how many bits the bitmap file q32x48.bin has

- 1536 bits

Step 2: Use Run Length function to compress the bitmap file q32x48.bin

- Java RunLength - < q32x48.bin > q32x48rle.bin
- Java BinaryDump < q32x48rle.bin
  - 1144 bits

Step 3: Calculate the compression ratio

- Compression ratio = 1144 / 1536
  - 0.7

Step 4: Perform the Steps 1 and 2 on the higher resolution bitmap file q64x96.bin

- Java BinaryDump < q64x96.bin
  - 6144 bits
- Java RunLength -< q64x96.bin > q64x96rle.bin
- Java BinaryDump < q64x96rle.bin
  - 2296 bits
- Compression ratio = 2296 / 6144 = 0.37

Step 4: Compare the compression ratio of the first bitmap image to this second compressed bitmap image. What do you think is the reason for this difference?

I believe this to be because, the larger file has more data to compress. It will have a larger initial size because it has more uncompressed data and this explains why the compression ratio is much less for the larger file.