**CAP6010: Multimedia Systems**

**Programming Assignment – JPEG Lossless Coder**

**Michael Lopez-Cruz**

**Z23346573**

**Index**

|  |  |
| --- | --- |
| Topic | Page |
| [Introduction](#Intro) | [1](#Intro) |
| [Program Results](#Results) | [2](#Results) |
| [Conclusion / Discussion](#Conclusion) | [28](#Conclusion) |
| [Program Code](#Code) | [30](#Code) |

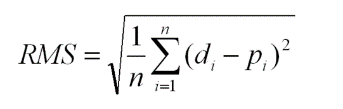
**Intro****duction**

In this assignment we will be writing a C# program to evaluate different encoding formulas for running a JPEG Lossless encoder that compresses its data for less storage usage. The lossless encoder and decoder can compress and decode a 16x16 image without losing any data. The program takes in a 16x16 pixel image in the form of a 2D array. The actual code of the program can be found at the end of this document and in the repository at <https://github.com/mlopezcruz2015/JPEG-Lossless-Encoder>. The image.txt file must be included in the directory where the program is being ran for the 16x16 image to be imported. The repo can be directly imported and should work by using Visual Studio.

The following results are directly from the program presented in a table; the structure for each iteration is the following:

* Title / Formula Used
* Original Image
* Coefficients after the predictor
* Compressed image in the form of a binary sequence
* Image after Huffman decoder
* Image after decompression
* Compression Ratio
* Bits/Pixel for compressed image
* RMS Error

Compression evaluation formulas used:

* Compression Ratio formula: 
* Bits / Pixel formula: **NC / 256**
* RMS Error:  where d = pixel from original image, p = pixel from decompressed image, n = 256

**Resul****ts**

|  |
| --- |
| **X = A** |
| Running First Formula X = A.  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** -**2** **0** -**1** **2** **1** **1** **0** **1** **0** -**3** **1** **2** **0** **0** **1**  **2** **0** **0** **0** **1** **1** -**3** **2** **1** -**1** **2** -**1** **1** **1** -**4** **2**  **2** -**1** **2** -**1** **1** -**4** **1** **2** **2** **1** -**1** **2** -**1** **0** -**2** **0**  **2** **1** -**3** **2** **1** **0** -**2** **0** **2** **1** -**1** -**1** **2** **1** -**3** **1**  **1** **1** -**1** -**1** **2** **1** -**3** **1** **3** -**1** **1** -**2** **0** -**1** **2** -**1**  **0** **2** **1** -**2** **0** -**1** **2** **1** **0** **2** -**1** **0** -**1** **1** **0** -**1**  **3** **2** -**2** **2** **0** -**1** **0** **0** -**2** -**1** -**1** **0** **0** **1** **1** **1**  **1** **0** **1** **0** **0** -**1** **0** **2** -**4** **1** -**2** **0** **1** -**1** **1** **0**  -**2** **3** **0** **0** **1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **3** **0** **0** **1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **3** **0** **0** **1** -**2** **1** -**1** -**1** -**2** **1** -**2** **2** **0** **0** **1**  **2** **1** **1** -**1** **1** -**2** **2** -**3** **0** -**1** -**2** **0** **1** **1** **0** **1**  -**1** **2** **2** **0** -**1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **2** **1** **2** -**2** -**2** **2** -**1** -**2** -**2** **0** -**1** **1** **2** -**1** **3**  -**1** **3** **0** **0** **1** -**2** **2** **1** -**3** -**1** -**1** -**2** **1** **3** -**2** **2**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **01011** **1** **011** **0100** **00** **00** **1** **00** **1** **0101011** **00** **0100** **1** **1** **00**  **0100** **1** **1** **1** **00** **00** **0101011** **0100** **00** **011** **0100** **011** **00** **00** **010101011** **0100**  **0100** **011** **0100** **011** **00** **010101011** **00** **0100** **0100** **00** **011** **0100** **011** **1** **01011** **1**  **0100** **00** **0101011** **0100** **00** **1** **01011** **1** **0100** **00** **011** **011** **0100** **00** **0101011** **00**  **00** **00** **011** **011** **0100** **00** **0101011** **00** **010100** **011** **00** **01011** **1** **011** **0100** **011**  **1** **0100** **00** **01011** **1** **011** **0100** **00** **1** **0100** **011** **1** **011** **00** **1** **011**  **010100** **0100** **01011** **0100** **1** **011** **1** **1** **01011** **011** **011** **1** **1** **00** **00** **00**  **00** **1** **00** **1** **1** **011** **1** **0100** **010101011** **00** **01011** **1** **00** **011** **00** **1**  **01011** **010100** **1** **1** **00** **01011** **0100** **01011** **011** **01011** **011** **1** **00** **00** **1** **00**  **1** **010100** **1** **1** **00** **01011** **0100** **01011** **011** **01011** **011** **1** **00** **00** **1** **00**  **1** **010100** **1** **1** **00** **01011** **00** **011** **011** **01011** **00** **01011** **0100** **1** **1** **00**  **0100** **00** **00** **011** **00** **01011** **0100** **0101011** **1** **011** **01011** **1** **00** **00** **1** **00**  **011** **0100** **0100** **1** **011** **01011** **0100** **01011** **011** **01011** **011** **1** **00** **00** **1** **00**  **1** **0100** **00** **0100** **01011** **01011** **0100** **011** **01011** **01011** **1** **011** **00** **0100** **011** **010100**  **011** **010100** **1** **1** **00** **01011** **0100** **00** **0101011** **011** **011** **01011** **00** **010100** **01011** **0100**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** -**2** **0** -**1** **2** **1** **1** **0** **1** **0** -**3** **1** **2** **0** **0** **1**  **2** **0** **0** **0** **1** **1** -**3** **2** **1** -**1** **2** -**1** **1** **1** -**4** **2**  **2** -**1** **2** -**1** **1** -**4** **1** **2** **2** **1** -**1** **2** -**1** **0** -**2** **0**  **2** **1** -**3** **2** **1** **0** -**2** **0** **2** **1** -**1** -**1** **2** **1** -**3** **1**  **1** **1** -**1** -**1** **2** **1** -**3** **1** **3** -**1** **1** -**2** **0** -**1** **2** -**1**  **0** **2** **1** -**2** **0** -**1** **2** **1** **0** **2** -**1** **0** -**1** **1** **0** -**1**  **3** **2** -**2** **2** **0** -**1** **0** **0** -**2** -**1** -**1** **0** **0** **1** **1** **1**  **1** **0** **1** **0** **0** -**1** **0** **2** -**4** **1** -**2** **0** **1** -**1** **1** **0**  -**2** **3** **0** **0** **1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **3** **0** **0** **1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **3** **0** **0** **1** -**2** **1** -**1** -**1** -**2** **1** -**2** **2** **0** **0** **1**  **2** **1** **1** -**1** **1** -**2** **2** -**3** **0** -**1** -**2** **0** **1** **1** **0** **1**  -**1** **2** **2** **0** -**1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **2** **1** **2** -**2** -**2** **2** -**1** -**2** -**2** **0** -**1** **1** **2** -**1** **3**  -**1** **3** **0** **0** **1** -**2** **2** **1** -**3** -**1** -**1** -**2** **1** **3** -**2** **2**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **776** = **2.64**  Average Bits/Pixel = **776** / **256** = **3.03**  RMS Error = **0** |
| **X = B** |
| Running Second Formula X = B.  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **3** **3** **1** **2** **2** **2** **1** **1** **0** -**1** -**2** -**1** -**3** -**2** **2**  **2** **4** **4** **5** **4** **4** **0** **2** **2** **1** **6** **4** **3** **4** **0** **1**  **2** **1** **3** **2** **2** -**3** **1** **1** **2** **4** **1** **4** **2** **1** **3** **1**  **2** **4** -**1** **2** **2** **6** **3** **1** **1** **1** **1** -**2** **1** **2** **1** **2**  **1** **1** **3** **0** **1** **2** **1** **2** **3** **1** **3** **2** **0** -**2** **3** **1**  **0** **1** **3** **2** **0** -**2** **3** **3** **0** **3** **1** **3** **2** **4** **2** **2**  **3** **3** **0** **4** **4** **4** **2** **1** -**1** -**4** -**4** -**4** -**3** -**3** -**2** **0**  **1** -**1** **2** **0** **0** **0** **0** **2** **0** **2** **1** **1** **2** **0** **0** -**1**  -**2** **1** **0** **0** **1** **0** **2** -**2** **1** -**2** -**1** -**1** -**1** **1** **0** **1**  **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0**  **0** **0** **0** **0** **0** **0** -**1** **0** **0** **0** **2** **0** **1** **0** **0** **0**  **2** **0** **1** **0** **0** **0** **1** -**1** **0** **1** -**2** **0** -**1** **0** **0** **0**  -**1** **0** **1** **2** **0** **0** **0** **1** **0** -**1** **0** **0** **0** **0** **0** **0**  **0** **0** -**1** **1** **0** **0** **0** **1** **0** **0** **1** **0** **0** **1** **0** **2**  -**1** **0** -**1** -**3** **0** **0** **0** **2** **1** **2** **1** **0** **0** **1** **0** -**1**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **010100** **010100** **00** **0100** **0100** **0100** **00** **00** **1** **011** **01011** **011** **0101011** **01011** **0100**  **0100** **01010100** **01010100** **0101010100** **01010100** **01010100** **1** **0100** **0100** **00010101010100** **01010100** **010100** **01010100** **1** **00**  **0100** **00** **010100** **0100** **0100** **0101011** **00** **00** **0100** **01010100** **00** **01010100** **0100** **00** **010100** **00**  **0100** **01010100** **011** **0100** **0100010101010100** **010100** **00** **00** **00** **00** **01011** **00** **0100** **00** **0100**  **00** **00** **010100** **1** **00** **0100** **00** **0100** **010100** **00** **010100** **0100** **1** **01011** **010100** **00**  **1** **00** **010100** **0100** **1** **01011** **010100** **010100** **1** **010100** **00** **010100** **0100** **01010100** **0100** **0100**  **010100** **010100** **1** **01010100** **01010100** **01010100** **0100** **00** **011** **010101011** **010101011** **010101011** **0101011** **0101011** **01011** **1**  **00** **011** **0100** **1** **1** **1** **1** **0100** **1** **0100** **00** **00** **0100** **1** **1** **011**  **01011** **00** **1** **1** **00** **1** **0100** **01011** **00** **01011** **011** **011** **011** **00** **1** **00**  **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1**  **1** **1** **1** **1** **1** **1** **011** **1** **1** **1** **0100** **1** **00** **1** **1** **1**  **0100** **1** **00** **1** **1** **1** **00** **011** **1** **00** **01011** **1** **011** **1** **1** **1**  **011** **1** **00** **0100** **1** **1** **1** **00** **1** **011** **1** **1** **1** **1** **1** **1**  **1** **1** **011** **00** **1** **1** **1** **00** **1** **1** **00** **1** **1** **00** **1** **0100**  **011** **1** **011** **0101011** **1** **1** **1** **0100** **00** **0100** **00** **1** **1** **00** **1** **011**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **3** **3** **1** **2** **2** **2** **1** **1** **0** -**1** -**2** -**1** -**3** -**2** **2**  **2** **4** **4** **5** **4** **4** **0** **2** **2** **1** **6** **4** **3** **4** **0** **1**  **2** **1** **3** **2** **2** -**3** **1** **1** **2** **4** **1** **4** **2** **1** **3** **1**  **2** **4** -**1** **2** **2** **6** **3** **1** **1** **1** **1** -**2** **1** **2** **1** **2**  **1** **1** **3** **0** **1** **2** **1** **2** **3** **1** **3** **2** **0** -**2** **3** **1**  **0** **1** **3** **2** **0** -**2** **3** **3** **0** **3** **1** **3** **2** **4** **2** **2**  **3** **3** **0** **4** **4** **4** **2** **1** -**1** -**4** -**4** -**4** -**3** -**3** -**2** **0**  **1** -**1** **2** **0** **0** **0** **0** **2** **0** **2** **1** **1** **2** **0** **0** -**1**  -**2** **1** **0** **0** **1** **0** **2** -**2** **1** -**2** -**1** -**1** -**1** **1** **0** **1**  **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0**  **0** **0** **0** **0** **0** **0** -**1** **0** **0** **0** **2** **0** **1** **0** **0** **0**  **2** **0** **1** **0** **0** **0** **1** -**1** **0** **1** -**2** **0** -**1** **0** **0** **0**  -**1** **0** **1** **2** **0** **0** **0** **1** **0** -**1** **0** **0** **0** **0** **0** **0**  **0** **0** -**1** **1** **0** **0** **0** **1** **0** **0** **1** **0** **0** **1** **0** **2**  -**1** **0** -**1** -**3** **0** **0** **0** **2** **1** **2** **1** **0** **0** **1** **0** -**1**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **789** = **2.6**  Average Bits/Pixel = **789** / **256** = **3.08**  RMS Error = **0** |
| **X = C** |
| Running Third Formula X = C.  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **3** **3** **2** **3** **3** **3** **2** **2** **1** -**3** **0** **0** -**1** -**3** -**1**  **2** **2** **4** **4** **6** **5** **1** **2** **3** **1** **3** **5** **5** **4** **0** **2**  **2** **1** **3** **2** **3** -**2** -**2** **3** **3** **3** **3** **3** **3** **2** -**1** **3**  **2** **3** **1** **1** **3** **2** **4** **3** **3** **2** **0** **0** **0** **2** -**1** **2**  **1** **2** **0** **2** **2** **2** -**1** **2** **5** **2** **2** **1** **2** -**1** **0** **2**  **0** **2** **2** **1** **2** -**1** **0** **4** **3** **2** **2** **1** **2** **3** **4** **1**  **3** **5** **1** **2** **4** **3** **4** **2** -**1** -**2** -**5** -**4** -**4** -**2** -**2** -**1**  **1** **1** **0** **2** **0** -**1** **0** **2** -**2** **1** **0** **1** **2** **1** **1** **0**  -**2** **1** **1** **0** **1** -**1** **2** **0** -**3** -**1** -**3** -**1** **0** **0** **1** **1**  **0** **3** **0** **0** **1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **3** **0** **0** **1** -**2** **1** -**2** -**1** -**2** **1** **0** **2** **1** **0** **1**  **2** **3** **1** **0** **1** -**2** **2** -**2** -**1** -**1** -**1** -**2** **1** **0** **0** **1**  -**1** **1** **2** **1** **1** -**2** **2** -**2** **0** -**2** -**2** **0** **1** **1** **0** **1**  **0** **2** **1** **1** -**1** -**2** **2** -**1** -**1** -**2** **0** **0** **1** **2** **0** **3**  -**1** **2** **0** -**1** -**2** -**2** **2** **1** -**1** **0** **1** -**1** **1** **3** -**1** **2**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **010100** **010100** **0100** **010100** **010100** **010100** **0100** **0100** **00** **0101011** **1** **1** **011** **0101011** **011**  **0100** **0100** **01010100** **01010100010101010100** **0101010100** **00** **0100** **010100** **00** **010100** **0101010100** **0101010100** **01010100** **1** **0100**  **0100** **00** **010100** **0100** **010100** **01011** **01011** **010100** **010100** **010100** **010100** **010100** **010100** **0100** **011** **010100**  **0100** **010100** **00** **00** **010100** **0100** **01010100** **010100** **010100** **0100** **1** **1** **1** **0100** **011** **0100**  **00** **0100** **1** **0100** **0100** **0100** **011** **0100** **0101010100** **0100** **0100** **00** **0100** **011** **1** **0100**  **1** **0100** **0100** **00** **0100** **011** **1** **01010100** **010100** **0100** **0100** **00** **0100** **010100** **01010100** **00**  **010100** **0101010100** **00** **0100** **01010100** **010100** **01010100** **0100** **011** **01011** **01010101011** **010101011** **010101011** **01011** **01011** **011**  **00** **00** **1** **0100** **1** **011** **1** **0100** **01011** **00** **1** **00** **0100** **00** **00** **1**  **01011** **00** **00** **1** **00** **011** **0100** **1** **0101011** **011** **0101011** **011** **1** **1** **00** **00**  **1** **010100** **1** **1** **00** **01011** **0100** **01011** **011** **01011** **011** **1** **00** **00** **1** **00**  **1** **010100** **1** **1** **00** **01011** **00** **01011** **011** **01011** **00** **1** **0100** **00** **1** **00**  **0100** **010100** **00** **1** **00** **01011** **0100** **01011** **011** **011** **011** **01011** **00** **1** **1** **00**  **011** **00** **0100** **00** **00** **01011** **0100** **01011** **1** **01011** **01011** **1** **00** **00** **1** **00**  **1** **0100** **00** **00** **011** **01011** **0100** **011** **011** **01011** **1** **1** **00** **0100** **1** **010100**  **011** **0100** **1** **011** **01011** **01011** **0100** **00** **011** **1** **00** **011** **00** **010100** **011** **0100**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **3** **3** **2** **3** **3** **3** **2** **2** **1** -**3** **0** **0** -**1** -**3** -**1**  **2** **2** **4** **4** **6** **5** **1** **2** **3** **1** **3** **5** **5** **4** **0** **2**  **2** **1** **3** **2** **3** -**2** -**2** **3** **3** **3** **3** **3** **3** **2** -**1** **3**  **2** **3** **1** **1** **3** **2** **4** **3** **3** **2** **0** **0** **0** **2** -**1** **2**  **1** **2** **0** **2** **2** **2** -**1** **2** **5** **2** **2** **1** **2** -**1** **0** **2**  **0** **2** **2** **1** **2** -**1** **0** **4** **3** **2** **2** **1** **2** **3** **4** **1**  **3** **5** **1** **2** **4** **3** **4** **2** -**1** -**2** -**5** -**4** -**4** -**2** -**2** -**1**  **1** **1** **0** **2** **0** -**1** **0** **2** -**2** **1** **0** **1** **2** **1** **1** **0**  -**2** **1** **1** **0** **1** -**1** **2** **0** -**3** -**1** -**3** -**1** **0** **0** **1** **1**  **0** **3** **0** **0** **1** -**2** **2** -**2** -**1** -**2** -**1** **0** **1** **1** **0** **1**  **0** **3** **0** **0** **1** -**2** **1** -**2** -**1** -**2** **1** **0** **2** **1** **0** **1**  **2** **3** **1** **0** **1** -**2** **2** -**2** -**1** -**1** -**1** -**2** **1** **0** **0** **1**  -**1** **1** **2** **1** **1** -**2** **2** -**2** **0** -**2** -**2** **0** **1** **1** **0** **1**  **0** **2** **1** **1** -**1** -**2** **2** -**1** -**1** -**2** **0** **0** **1** **2** **0** **3**  -**1** **2** **0** -**1** -**2** -**2** **2** **1** -**1** **0** **1** -**1** **1** **3** -**1** **2**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **948** = **2.16**  Average Bits/Pixel = **948** / **256** = **3.7**  RMS Error = **0** |
| **X = A + B - C** |
| Running Fourth Formula X = A + B - C.  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** -**2** **0** -**2** **1** **0** **0** -**1** **0** -**1** -**1** -**1** **1** -**2** **1** **4**  **2** **2** **0** **1** -**1** **0** -**4** **2** **0** -**1** **5** -**2** -**1** **1** -**4** **1**  **2** -**1** **2** -**1** **0** -**5** **4** **0** **1** **2** -**3** **3** -**2** -**1** **2** -**2**  **2** **2** -**5** **3** **0** **4** -**3** -**2** **0** **0** **0** -**3** **3** **1** -**1** **1**  **1** **0** **2** -**3** **1** **1** -**1** **1** **1** -**2** **2** -**1** -**2** -**2** **5** -**2**  **0** **1** **2** -**1** -**2** -**2** **5** **0** -**3** **3** -**2** **2** -**1** **2** -**2** **0**  **3** **0** -**3** **4** **0** **0** -**2** -**1** -**2** -**3** **0** **0** **1** **0** **1** **2**  **1** -**2** **3** -**2** **0** **0** **0** **2** -**2** **2** -**1** **0** **1** -**2** **0** -**1**  -**2** **3** -**1** **0** **1** -**1** **2** -**4** **3** -**3** **1** **0** **0** **2** -**1** **1**  **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0**  **0** **0** **0** **0** **0** **0** -**1** **1** **0** **0** **2** -**2** **1** -**1** **0** **0**  **2** -**2** **1** -**1** **0** **0** **1** -**2** **1** **1** -**3** **2** -**1** **1** **0** **0**  -**1** **1** **1** **1** -**2** **0** **0** **1** -**1** -**1** **1** **0** **0** **0** **0** **0**  **0** **0** -**1** **2** -**1** **0** **0** **1** -**1** **0** **1** -**1** **0** **1** -**1** **2**  -**1** **1** -**1** -**2** **3** **0** **0** **2** -**1** **1** -**1** -**1** **0** **1** -**1** -**1**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **01011** **1** **01011** **00** **1** **1** **011** **1** **011** **011** **011** **00** **01011** **00** **01010100**  **0100** **0100** **1** **00** **011** **1** **010101011** **0100** **1** **011** **0101010100** **01011** **011** **00** **010101011** **00**  **0100** **011** **0100** **011** **101010101011** **01010100** **1** **00** **0100** **0101011** **010100** **01011** **011** **0100** **01011**  **0100** **010001010101011** **010100** **1** **01010100** **0101011** **01011** **1** **1** **1** **0101011** **010100** **00** **011** **00**  **00** **1** **0100** **0101011** **00** **00** **011** **00** **00** **01011** **0100** **011** **01011** **01011** **0101010100** **01011**  **1** **00** **0100** **011** **01011** **01011** **0101010100** **1** **0101011** **010100** **01011** **0100** **011** **0100** **01011** **1**  **010100** **1** **0101011** **01010100** **1** **1** **01011** **011** **01011** **0101011** **1** **1** **00** **1** **00** **0100**  **00** **01011** **010100** **01011** **1** **1** **1** **0100** **01011** **0100** **011** **1** **00** **01011** **1** **011**  **01011** **010100** **011** **1** **00** **011** **0100** **010101011** **010100** **0101011** **00** **1** **1** **0100** **011** **00**  **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1**  **1** **1** **1** **1** **1** **1** **011** **00** **1** **1** **0100** **01011** **00** **011** **1** **1**  **0100** **01011** **00** **011** **1** **1** **00** **01011** **00** **00** **0101011** **0100** **011** **00** **1** **1**  **011** **00** **00** **00** **01011** **1** **1** **00** **011** **011** **00** **1** **1** **1** **1** **1**  **1** **1** **011** **0100** **011** **1** **1** **00** **011** **1** **00** **011** **1** **00** **011** **0100**  **011** **00** **011** **01011** **010100** **1** **1** **0100** **011** **00** **011** **011** **1** **00** **011** **011**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** -**2** **0** -**2** **1** **0** **0** -**1** **0** -**1** -**1** -**1** **1** -**2** **1** **4**  **2** **2** **0** **1** -**1** **0** -**4** **2** **0** -**1** **5** -**2** -**1** **1** -**4** **1**  **2** -**1** **2** -**1** **0** -**5** **4** **0** **1** **2** -**3** **3** -**2** -**1** **2** -**2**  **2** **2** -**5** **3** **0** **4** -**3** -**2** **0** **0** **0** -**3** **3** **1** -**1** **1**  **1** **0** **2** -**3** **1** **1** -**1** **1** **1** -**2** **2** -**1** -**2** -**2** **5** -**2**  **0** **1** **2** -**1** -**2** -**2** **5** **0** -**3** **3** -**2** **2** -**1** **2** -**2** **0**  **3** **0** -**3** **4** **0** **0** -**2** -**1** -**2** -**3** **0** **0** **1** **0** **1** **2**  **1** -**2** **3** -**2** **0** **0** **0** **2** -**2** **2** -**1** **0** **1** -**2** **0** -**1**  -**2** **3** -**1** **0** **1** -**1** **2** -**4** **3** -**3** **1** **0** **0** **2** -**1** **1**  **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0**  **0** **0** **0** **0** **0** **0** -**1** **1** **0** **0** **2** -**2** **1** -**1** **0** **0**  **2** -**2** **1** -**1** **0** **0** **1** -**2** **1** **1** -**3** **2** -**1** **1** **0** **0**  -**1** **1** **1** **1** -**2** **0** **0** **1** -**1** -**1** **1** **0** **0** **0** **0** **0**  **0** **0** -**1** **2** -**1** **0** **0** **1** -**1** **0** **1** -**1** **0** **1** -**1** **2**  -**1** **1** -**1** -**2** **3** **0** **0** **2** -**1** **1** -**1** -**1** **0** **1** -**1** -**1**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **798** = **2.57**  Average Bits/Pixel = **798** / **256** = **3.12**  RMS Error = **0** |
| **X = A + (B - C)/2** |
| Running Fifth Formula X = A + (B - C)/**2.**  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** -**2** **0** -**1** **2** **1** **1** **0** **1** **0** -**2** **0** **2** -**1** **0** **2**  **2** **1** **0** **0** **0** **1** -**3** **2** **1** -**1** **3** -**1** **0** **1** -**4** **2**  **2** -**1** **2** -**1** **1** -**4** **2** **1** **2** **1** -**2** **2** -**1** **0** **0** -**1**  **2** **1** -**4** **2** **1** **2** -**2** -**1** **1** **1** -**1** -**2** **2** **1** -**2** **1**  **1** **1** **0** -**2** **2** **1** -**2** **1** **2** -**1** **1** -**2** -**1** -**1** **3** -**1**  **0** **2** **1** -**2** -**1** -**1** **3** **1** -**1** **2** -**1** **1** -**1** **1** -**1** -**1**  **3** **1** -**2** **3** **0** -**1** -**1** **0** -**2** -**2** -**1** **0** **0** **1** **1** **1**  **1** -**1** **2** -**1** **0** -**1** **0** **2** -**3** **1** -**2** **0** **1** -**1** **1** **0**  -**2** **3** **0** **0** **1** -**2** **2** -**3** **1** -**2** **0** **0** **1** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **1** -**1** -**1** -**1** -**1** **0** **1** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **0** **0** -**1** -**1** **1** -**2** **2** **0** **0** **1**  **2** **0** **1** -**1** **1** -**1** **2** -**3** **0** **0** -**2** **1** **0** **1** **0** **1**  -**1** **2** **2** **0** -**1** -**1** **1** -**1** -**1** -**2** **0** **0** **1** **1** **0** **1**  **0** **1** **0** **2** -**2** -**1** **1** **0** -**2** -**1** **0** -**1** **1** **2** -**1** **3**  -**1** **2** **0** -**1** **2** -**1** **1** **1** -**2** **0** -**1** -**2** **1** **2** -**2** **1**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **01011** **1** **011** **0100** **00** **00** **1** **00** **1** **01011** **1** **0100** **011** **1** **0100**  **0100** **00** **1** **1** **1** **00** **0101011** **0100** **00** **011** **010100** **011** **1** **00** **010101011** **0100**  **0100** **011** **0100** **011** **00** **010101011** **0100** **00** **0100** **00** **01011** **0100** **011** **1** **1** **011**  **0100** **00** **010101011** **0100** **00** **0100** **01011** **011** **00** **00** **011** **01011** **0100** **00** **01011** **00**  **00** **00** **1** **01011** **0100** **00** **01011** **00** **0100** **011** **00** **01011** **011** **011** **010100** **011**  **1** **0100** **00** **01011** **011** **011** **010100** **00** **011** **0100** **011** **00** **011** **00** **011** **011**  **010100** **00** **01011** **010100** **1** **011** **011** **1** **01011** **01011** **011** **1** **1** **00** **00** **00**  **00** **011** **0100** **011** **1** **011** **1** **0100** **0101011** **00** **01011** **1** **00** **011** **00** **1**  **01011** **010100** **1** **1** **00** **01011** **0100** **0101011** **00** **01011** **1** **1** **00** **00** **1** **00**  **1** **0100** **1** **1** **00** **011** **00** **011** **011** **011** **011** **1** **00** **00** **1** **00**  **1** **0100** **1** **1** **00** **011** **1** **1** **011** **011** **00** **01011** **0100** **1** **1** **00**  **0100** **1** **00** **011** **00** **011** **0100** **0101011** **1** **1** **01011** **00** **1** **00** **1** **00**  **011** **0100** **0100** **1** **011** **011** **00** **011** **011** **01011** **1** **1** **00** **00** **1** **00**  **1** **00** **1** **0100** **01011** **011** **00** **1** **01011** **011** **1** **011** **00** **0100** **011** **010100**  **011** **0100** **1** **011** **0100** **011** **00** **00** **01011** **1** **011** **01011** **00** **0100** **01011** **00**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** -**2** **0** -**1** **2** **1** **1** **0** **1** **0** -**2** **0** **2** -**1** **0** **2**  **2** **1** **0** **0** **0** **1** -**3** **2** **1** -**1** **3** -**1** **0** **1** -**4** **2**  **2** -**1** **2** -**1** **1** -**4** **2** **1** **2** **1** -**2** **2** -**1** **0** **0** -**1**  **2** **1** -**4** **2** **1** **2** -**2** -**1** **1** **1** -**1** -**2** **2** **1** -**2** **1**  **1** **1** **0** -**2** **2** **1** -**2** **1** **2** -**1** **1** -**2** -**1** -**1** **3** -**1**  **0** **2** **1** -**2** -**1** -**1** **3** **1** -**1** **2** -**1** **1** -**1** **1** -**1** -**1**  **3** **1** -**2** **3** **0** -**1** -**1** **0** -**2** -**2** -**1** **0** **0** **1** **1** **1**  **1** -**1** **2** -**1** **0** -**1** **0** **2** -**3** **1** -**2** **0** **1** -**1** **1** **0**  -**2** **3** **0** **0** **1** -**2** **2** -**3** **1** -**2** **0** **0** **1** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **1** -**1** -**1** -**1** -**1** **0** **1** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **0** **0** -**1** -**1** **1** -**2** **2** **0** **0** **1**  **2** **0** **1** -**1** **1** -**1** **2** -**3** **0** **0** -**2** **1** **0** **1** **0** **1**  -**1** **2** **2** **0** -**1** -**1** **1** -**1** -**1** -**2** **0** **0** **1** **1** **0** **1**  **0** **1** **0** **2** -**2** -**1** **1** **0** -**2** -**1** **0** -**1** **1** **2** -**1** **3**  -**1** **2** **0** -**1** **2** -**1** **1** **1** -**2** **0** -**1** -**2** **1** **2** -**2** **1**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **746** = **2.75**  Average Bits/Pixel = **746** / **256** = **2.91**  RMS Error = **0** |
| **X = B + (A - C)/2** |
| Running Sixth Formula X = B + (A - C)/**2.**  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **1** **2** **0** **2** **1** **1** **0** **1** **0** -**1** -**2** **0** -**3** -**1** **3**  **2** **3** **2** **3** **2** **2** -**2** **2** **1** **0** **6** **1** **1** **3** -**2** **1**  **2** **0** **3** **1** **1** -**4** **2** **1** **2** **3** -**1** **4** **0** **0** **3** **0**  **2** **3** -**3** **2** **1** **5** **0** **0** **1** **1** **1** -**2** **2** **2** **0** **2**  **1** **1** **3** -**1** **1** **2** **0** **2** **2** **0** **3** **1** -**1** -**2** **4** **0**  **0** **1** **3** **1** -**1** -**2** **4** **2** -**1** **3** **0** **3** **1** **3** **0** **1**  **3** **2** -**1** **4** **2** **2** **0** **0** -**1** -**4** -**2** -**2** -**1** -**2** -**1** **1**  **1** -**1** **2** -**1** **0** **0** **0** **2** -**1** **2** **0** **1** **2** -**1** **0** -**1**  -**2** **2** **0** **0** **1** **0** **2** -**3** **2** -**2** **0** -**1** -**1** **1** **0** **1**  **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0**  **0** **0** **0** **0** **0** **0** -**1** **0** **0** **0** **2** -**1** **1** **0** **0** **0**  **2** -**1** **1** **0** **0** **0** **1** -**1** **0** **1** -**2** **1** -**1** **0** **0** **0**  -**1** **0** **1** **2** -**1** **0** **0** **1** **0** -**1** **0** **0** **0** **0** **0** **0**  **0** **0** -**1** **1** **0** **0** **0** **1** **0** **0** **1** **0** **0** **1** **0** **2**  -**1** **0** -**1** -**3** **1** **0** **0** **2** **0** **2** **0** **0** **0** **1** **0** -**1**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **00** **0100** **1** **0100** **00** **00** **1** **00** **1** **011** **01011** **1** **0101011** **011** **010100**  **0100** **010100** **0100** **010100** **0100** **0100** **01011** **0100** **00** **1010101010100** **00** **00** **010100** **01011** **00**  **0100** **1** **010100** **00** **00** **010101011** **0100** **00** **0100** **010100** **011** **01010100** **1** **1** **010100** **1**  **0100** **010100** **0101011** **0100** **00** **0101010100** **1** **1** **00** **00** **00** **01011** **0100** **0100** **1** **0100**  **00** **00** **010100** **011** **00** **0100** **1** **0100** **0100** **1** **010100** **00** **011** **01011** **01010100** **1**  **1** **00** **010100** **00** **011** **01011** **01010100** **0100** **011** **010100** **1** **010100** **00** **010100** **1** **00**  **010100** **0100** **011** **01010100** **0100** **0100** **1** **1** **011** **010101011** **01011** **01011** **011** **01011** **011** **00**  **00** **011** **0100** **011** **1** **1** **1** **0100** **011** **0100** **1** **00** **0100** **011** **1** **011**  **01011** **0100** **1** **1** **00** **1** **0100** **0101011** **0100** **01011** **1** **011** **011** **00** **1** **00**  **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1** **1**  **1** **1** **1** **1** **1** **1** **011** **1** **1** **1** **0100** **011** **00** **1** **1** **1**  **0100** **011** **00** **1** **1** **1** **00** **011** **1** **00** **01011** **00** **011** **1** **1** **1**  **011** **1** **00** **0100** **011** **1** **1** **00** **1** **011** **1** **1** **1** **1** **1** **1**  **1** **1** **011** **00** **1** **1** **1** **00** **1** **1** **00** **1** **1** **00** **1** **0100**  **011** **1** **011** **0101011** **00** **1** **1** **0100** **1** **0100** **1** **1** **1** **00** **1** **011**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **1** **2** **0** **2** **1** **1** **0** **1** **0** -**1** -**2** **0** -**3** -**1** **3**  **2** **3** **2** **3** **2** **2** -**2** **2** **1** **0** **6** **1** **1** **3** -**2** **1**  **2** **0** **3** **1** **1** -**4** **2** **1** **2** **3** -**1** **4** **0** **0** **3** **0**  **2** **3** -**3** **2** **1** **5** **0** **0** **1** **1** **1** -**2** **2** **2** **0** **2**  **1** **1** **3** -**1** **1** **2** **0** **2** **2** **0** **3** **1** -**1** -**2** **4** **0**  **0** **1** **3** **1** -**1** -**2** **4** **2** -**1** **3** **0** **3** **1** **3** **0** **1**  **3** **2** -**1** **4** **2** **2** **0** **0** -**1** -**4** -**2** -**2** -**1** -**2** -**1** **1**  **1** -**1** **2** -**1** **0** **0** **0** **2** -**1** **2** **0** **1** **2** -**1** **0** -**1**  -**2** **2** **0** **0** **1** **0** **2** -**3** **2** -**2** **0** -**1** -**1** **1** **0** **1**  **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0** **0**  **0** **0** **0** **0** **0** **0** -**1** **0** **0** **0** **2** -**1** **1** **0** **0** **0**  **2** -**1** **1** **0** **0** **0** **1** -**1** **0** **1** -**2** **1** -**1** **0** **0** **0**  -**1** **0** **1** **2** -**1** **0** **0** **1** **0** -**1** **0** **0** **0** **0** **0** **0**  **0** **0** -**1** **1** **0** **0** **0** **1** **0** **0** **1** **0** **0** **1** **0** **2**  -**1** **0** -**1** -**3** **1** **0** **0** **2** **0** **2** **0** **0** **0** **1** **0** -**1**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **715** = **2.86**  Average Bits/Pixel = **715** / **256** = **2.79**  RMS Error = **0** |
| **X = (A + B)/2** |
| Running Seventh Formula X = (A + B)/**2.**  Original **16**x16 image:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Coefficients after predictor:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **1** **2** **0** **2** **2** **2** **1** **1** **0** -**2** **0** **1** -**1** -**1** **2**  **2** **2** **2** **3** **3** **3** -**1** **2** **2** **0** **4** **2** **2** **3** -**2** **2**  **2** **0** **3** **1** **2** -**3** **1** **2** **2** **3** **0** **3** **1** **1** **1** **1**  **2** **3** -**2** **2** **2** **3** **1** **1** **2** **1** **0** -**1** **2** **2** -**1** **2**  **1** **1** **1** **0** **2** **2** -**1** **2** **3** **0** **2** **0** **0** -**1** **3** **0**  **0** **2** **2** **0** **0** -**1** **3** **2** **0** **3** **0** **2** **1** **3** **1** **1**  **3** **3** -**1** **3** **2** **2** **1** **1** -**1** -**2** -**2** -**2** -**1** -**1** **0** **1**  **1** **0** **2** **0** **0** **0** **0** **2** -**2** **2** **0** **1** **2** **0** **1** **0**  -**2** **2** **0** **0** **1** -**1** **2** -**2** **0** -**2** -**1** **0** **0** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **1** -**1** **0** -**1** **0** **0** **1** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **0** **0** **0** -**1** **2** -**1** **2** **0** **0** **1**  **2** **1** **1** **0** **1** -**1** **2** -**2** **0** **0** -**2** **0** **0** **1** **0** **1**  -**1** **1** **2** **1** **0** -**1** **1** **0** **0** -**1** **0** **0** **1** **1** **0** **1**  **0** **1** **0** **2** -**1** -**1** **1** **0** -**1** -**1** **1** **0** **1** **2** **0** **3**  -**1** **2** **0** -**1** **1** -**1** **1** **2** -**1** **1** **0** -**1** **1** **2** -**1** **1**  Compressed Image **as** a binary sequence:  **1011000** **1** **1** **00** **00** **00** **00** **00** **00** **00** **01011** **0100** **00** **0100** **011** **0101011**  **0101010100** **00** **0100** **1** **0100** **0100** **0100** **00** **00** **1** **01011** **1** **00** **011** **011** **0100**  **0100** **0100** **0100** **010100** **010100** **010100** **011** **0100** **0100** **1** **01010100** **0100** **0100** **010100** **01011** **0100**  **0100** **1** **010100** **00** **0100** **0101011** **00** **0100** **0100** **010100** **1** **010100** **00** **00** **00** **00**  **0100** **010100** **01011** **0100** **0100** **010100** **00** **00** **0100** **00** **1** **011** **0100** **0100** **011** **0100**  **00** **00** **00** **1** **0100** **0100** **011** **0100** **010100** **1** **0100** **1** **1** **011** **010100** **1**  **1** **0100** **0100** **1** **1** **011** **010100** **0100** **1** **010100** **1** **0100** **00** **010100** **00** **00**  **010100** **010100** **011** **010100** **0100** **0100** **00** **00** **011** **01011** **01011** **01011** **011** **011** **1** **00**  **00** **1** **0100** **1** **1** **1** **1** **0100** **01011** **0100** **1** **00** **0100** **1** **00** **1**  **01011** **0100** **1** **1** **00** **011** **0100** **01011** **1** **01011** **011** **1** **1** **00** **1** **00**  **1** **0100** **1** **1** **00** **011** **00** **011** **1** **011** **1** **1** **00** **00** **1** **00**  **1** **0100** **1** **1** **00** **011** **1** **1** **1** **011** **0100** **011** **0100** **1** **1** **00**  **0100** **00** **00** **1** **00** **011** **0100** **01011** **1** **1** **01011** **1** **1** **00** **1** **00**  **011** **00** **0100** **00** **1** **011** **00** **1** **1** **011** **1** **1** **00** **00** **1** **00**  **1** **00** **1** **0100** **011** **011** **00** **1** **011** **011** **00** **1** **00** **0100** **1** **010100**  **011** **0100** **1** **011** **00** **011** **00** **0100** **011** **00** **1** **011** **00** **0100** **011** **00**  Image after Huffman Decoder:  **88** **0** **0** **1** **1** **1** **1** **1** **1** **1** -**2** **2** **1** **2** -**1** -**3**  **5** **1** **2** **0** **2** **2** **2** **1** **1** **0** -**2** **0** **1** -**1** -**1** **2**  **2** **2** **2** **3** **3** **3** -**1** **2** **2** **0** **4** **2** **2** **3** -**2** **2**  **2** **0** **3** **1** **2** -**3** **1** **2** **2** **3** **0** **3** **1** **1** **1** **1**  **2** **3** -**2** **2** **2** **3** **1** **1** **2** **1** **0** -**1** **2** **2** -**1** **2**  **1** **1** **1** **0** **2** **2** -**1** **2** **3** **0** **2** **0** **0** -**1** **3** **0**  **0** **2** **2** **0** **0** -**1** **3** **2** **0** **3** **0** **2** **1** **3** **1** **1**  **3** **3** -**1** **3** **2** **2** **1** **1** -**1** -**2** -**2** -**2** -**1** -**1** **0** **1**  **1** **0** **2** **0** **0** **0** **0** **2** -**2** **2** **0** **1** **2** **0** **1** **0**  -**2** **2** **0** **0** **1** -**1** **2** -**2** **0** -**2** -**1** **0** **0** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **1** -**1** **0** -**1** **0** **0** **1** **1** **0** **1**  **0** **2** **0** **0** **1** -**1** **0** **0** **0** -**1** **2** -**1** **2** **0** **0** **1**  **2** **1** **1** **0** **1** -**1** **2** -**2** **0** **0** -**2** **0** **0** **1** **0** **1**  -**1** **1** **2** **1** **0** -**1** **1** **0** **0** -**1** **0** **0** **1** **1** **0** **1**  **0** **1** **0** **2** -**1** -**1** **1** **0** -**1** -**1** **1** **0** **1** **2** **0** **3**  -**1** **2** **0** -**1** **1** -**1** **1** **2** -**1** **1** **0** -**1** **1** **2** -**1** **1**  Image after Decompression:  **88** **88** **88** **89** **90** **91** **92** **93** **94** **95** **93** **95** **96** **98** **97** **94**  **93** **91** **91** **90** **92** **93** **94** **94** **95** **95** **92** **93** **95** **95** **95** **96**  **95** **95** **95** **95** **96** **97** **94** **96** **97** **96** **98** **97** **98** **99** **95** **97**  **97** **96** **98** **97** **98** **94** **95** **97** **99** **100** **99** **101** **100** **100** **98** **98**  **99** **100** **97** **99** **100** **100** **98** **98** **100** **101** **100** **99** **101** **102** **99** **100**  **100** **101** **100** **99** **101** **102** **99** **100** **103** **102** **103** **101** **101** **100** **102** **101**  **100** **102** **103** **101** **101** **100** **102** **103** **103** **105** **104** **104** **103** **104** **104** **103**  **103** **105** **103** **105** **105** **104** **104** **104** **102** **101** **100** **100** **100** **101** **102** **103**  **104** **104** **105** **105** **105** **104** **104** **106** **102** **103** **101** **101** **102** **101** **102** **102**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **102** **105** **105** **105** **106** **104** **105** **104** **103** **101** **102** **100** **102** **102** **102** **103**  **104** **105** **106** **105** **106** **104** **106** **103** **103** **102** **100** **100** **101** **102** **102** **103**  **103** **105** **107** **107** **106** **104** **106** **104** **103** **101** **100** **100** **101** **102** **102** **103**  **103** **105** **106** **108** **106** **104** **106** **105** **103** **101** **101** **100** **101** **103** **102** **105**  **102** **105** **105** **105** **106** **104** **106** **107** **104** **103** **102** **100** **101** **104** **102** **104**  Compression Ratio = **2048** / **727** = **2.82**  Average Bits/Pixel = **727** / **256** = **2.84**  RMS Error = **0** |

**Conclusion /** **Discussion**

For al instances ran, the RMS resulted in 0, which means that our program successfully encoded and decoded our image without any loss of data. The following are the compression ratios and bits per pixel for each formula:

1. X = A

Compression Ratio = **2048** / **776** = **2.64**

Average Bits/Pixel = **776** / **256** = **3.03**

1. X = B

Compression Ratio = **2048** / **789** = **2.6**

Average Bits/Pixel = **789** / **256** = **3.08**

1. X = C

Compression Ratio = **2048** / **948** = **2.16**

Average Bits/Pixel = **948** / **256** = **3.7**

1. X = A + B – C

Compression Ratio = **2048** / **798** = **2.57**

Average Bits/Pixel = **798** / **256** = **3.12**

1. X = A + (B – C)/2

Compression Ratio = **2048** / **746** = **2.75**

Average Bits/Pixel = **746** / **256** = **2.91**

1. X = B + (A - C)/2

Compression Ratio = **2048** / **715** = **2.86**

Average Bits/Pixel = **715** / **256** = **2.79**

1. X = (A + B)/2

Compression Ratio = **2048** / **727** = **2.82**

Average Bits/Pixel = **727** / **256** = **2.84**

Average compression ratio = 2.63

Average Bits/Pixel for all formulas = 3.07

We are looking for the highest compression ratio and the lowest average bits per pixel to determine which is the best formula to use for lossless encoding, at least for this image. In this case, the 6th formula X = B + (A - C)/2 gave us the best numbers with a compression ratio of 2.86 and 2.79 average bits / pixel. All formulas used gave us an exceptional compression rate averaging 2.63, which further proves that JPEG lossless encoding gives us great benefit in real world use.

**Program C****ode**

|  |
| --- |
| **using** **System**;  **using** **System.Collections.Generic**;  **using** **System.IO**;  **namespace** **JPEGLosslessEncoder**  {  **class** **Program**  {  //Use the following Huffman table for entropy encoding  **private** **static** **string** **HuffmanTableEncoder**(**int** key)  {  **switch** (key)  {  **case** **0**:  **return** "1";  **case** **1**:  **return** "00";  **case** -**1**:  **return** "011";  **case** **2**:  **return** "0100";  **case** -**2**:  **return** "01011";  **case** **3**:  **return** "010100";  **case** -**3**:  **return** "0101011";  **case** **4**:  **return** "01010100";  **case** -**4**:  **return** "010101011";  **case** **5**:  **return** "0101010100";  **case** -**5**:  **return** "01010101011";  **case** **6**:  **return** "010101010100";  **case** -**6**:  **return** "0101010101011";  **default**:  **return** Convert.ToString(key, **2**);  }  }  //Use the following Huffman table for entropy decoding  **private** **static** **int** **HuffmanTableDecoder**(**string** b)  {  **switch** (b)  {  **case** "1":  **return** **0**;  **case** "00":  **return** **1**;  **case** "011":  **return** -**1**;  **case** "0100":  **return** **2**;  **case** "01011":  **return** -**2**;  **case** "010100":  **return** **3**;  **case** "0101011":  **return** -**3**;  **case** "01010100":  **return** **4**;  **case** "010101011":  **return** -**4**;  **case** "0101010100":  **return** **5**;  **case** "01010101011":  **return** -**5**;  **case** "010101010100":  **return** **6**;  **case** "0101010101011":  **return** -**6**;  **default**:  **var** x = b.PadLeft(**8**, '0');  **return** Convert.ToInt32(x, **2**);  }  }  **static** **void** **Main**(**string**[] args)  {  **try**  {  //string path = Directory.GetCurrentDirectory();  **string** text = System.IO.File.ReadAllText(@"image.txt").Replace("\r\n", " ");  **var** formattedText = text.Split(" ");  //Use a 2d array to store original image.  **int**[,] originalImage = **new** **int**[**16**, **16**];  **for** (**int** i = **0**; i < formattedText.Length; i++)  {  originalImage[i / **16**, i % **16**] = **int**.Parse(formattedText[i]);  }  **if** (originalImage.Length != **256**)  {  Console.WriteLine("Program Stopped, bad data. Image must be 16x16.");  **return**;  }  **for** (**int** i = **1**; i < **8**; i++)  {  **var** coefficients = RunEncoderFormula(i, originalImage);  RunEncoderAndDecoder(i, coefficients);  RunCalculations(i, originalImage, coefficients);  Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  Console.WriteLine(Environment.NewLine);  }  }  **catch** (Exception ex)  {  Console.WriteLine($"Error running program - {ex.Message}");  }  }  /// <summary>  /// Returns Encoded array using the index i as the selector for which formula to use for the encoding.  /// </summary>  /// <param name="i"></param>  **private** **static** **int**[,] RunEncoderFormula(**int** i, **int**[,] array)  {  **switch** (i)  {  **case** **1**:  **return** **FirstFormula**(array);  **case** **2**:  **return** **SecondFormula**(array);  **case** **3**:  **return** **ThirdFormula**(array);  **case** **4**:  **return** **FourthFormula**(array);  **case** **5**:  **return** **FifthFormula**(array);  **case** **6**:  **return** **SixthFormula**(array);  **case** **7**:  **return** **SeventhFormula**(array);  **default**:  **return** **null**;  }  }  /// <summary>  /// X = A  /// First row always X = A, First Column always X = B  /// </summary>  /// <param name="array"></param>  **static** **int**[,] FirstFormula(**int**[,] array)  {  Console.WriteLine("Running First Formula X = A.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **var** coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  coefficients[i, j] = DoXEqualsA(array, i, j);  }  }  **return** coefficients;  }  /// <summary>  /// X = B  /// First row always X = A, First Column always X = B  /// </summary>  /// <param name="array"></param>  **static** **int**[,] SecondFormula(**int**[,] array)  {  Console.WriteLine("Running Second Formula X = B.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **var** coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  coefficients[i, j] = DoXEqualsB(array, i, j);  }  }  **return** coefficients;  }  /// <summary>  /// X = C  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **int**[,] ThirdFormula(**int**[,] array)  {  Console.WriteLine("Running Third Formula X = C.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **int**[,] coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  coefficients[i, j] = array[i, j] - array[i - **1**, j - **1**];  }  }  **return** coefficients;  }  /// <summary>  /// X = A + B - C  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **int**[,] FourthFormula(**int**[,] array)  {  Console.WriteLine("Running Fourth Formula X = A + B - C.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **int**[,] coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  coefficients[i, j] = array[i, j] - (array[i, j - **1**] + array[i - **1**, j] - array[i - **1**, j - **1**]);  }  }  **return** coefficients;  }  /// <summary>  /// X = A + (B-C)/2  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **int**[,] FifthFormula(**int**[,] array)  {  Console.WriteLine("Running Fifth Formula X = A + (B - C)/2.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **int**[,] coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  **var** x = (**int**)Math.Ceiling((**double**)((array[i - **1**, j] - array[i - **1**, j - **1**]) / **2**));  coefficients[i, j] = array[i, j] - (array[i, j - **1**] + x);  }  }  **return** coefficients;  }  /// <summary>  /// X = B + (A-C)/2  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **int**[,] SixthFormula(**int**[,] array)  {  Console.WriteLine("Running Sixth Formula X = B + (A - C)/2.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **int**[,] coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  **var** x = (**int**)Math.Ceiling((**double**)((array[i, j - **1**] - array[i - **1**, j - **1**]) / **2**));  coefficients[i, j] = array[i, j] - (array[i - **1**, j] + x);  }  }  **return** coefficients;  }  /// <summary>  /// X = (A + B)/2  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **int**[,] SeventhFormula(**int**[,] array)  {  Console.WriteLine("Running Seventh Formula X = (A + B)/2.");  Console.WriteLine("Original 16x16 image:");  print2DArray(array);  **int**[,] coefficients = InitializeCoefficientsArray(array);  **for** (**int** i = **1**; i < **16**; i++)  {  **for** (**int** j = **1**; j < **16**; j++)  {  **var** x = (**int**)Math.Ceiling((**double**)((array[i, j - **1**] + array[i - **1**, j]) / **2**));  coefficients[i, j] = array[i, j] - x;  }  }  **return** coefficients;  }  /// <summary>  /// Runs the formula X = A to encode the image  /// </summary>  /// <param name="array"></param>  /// <param name="i"></param>  /// <param name="j"></param>  /// <returns></returns>  **static** **int** **DoXEqualsA**(**int**[,] array, **int** i, **int** j)  {  **return** array[i, j] - array[i, j - **1**];  }  /// <summary>  /// Runs the formula X = B to encode the image  /// </summary>  /// <param name="array"></param>  /// <param name="i"></param>  /// <param name="j"></param>  /// <returns></returns>  **static** **int** **DoXEqualsB**(**int**[,] array, **int** i, **int** j)  {  **return** array[i, j] - array[i - **1**, j];  }  /// <summary>  /// Prints The Following:  /// The coefficients after the predictor  /// The compressed image in the form of binary sequence.  /// The image after Huffman decoder  /// The image after decompression (the sequence of numbers)  /// </summary>  /// <param name="coefficients"></param>  **private** **static** **void** **RunEncoderAndDecoder**(**int** i, **int**[,] coefficients)  {  Console.WriteLine("Coefficients after predictor:");  print2DArray(coefficients);  Console.WriteLine("Compressed Image as a binary sequence:");  **var** encodedArray = EncodeBinarySequence(coefficients);  print2DArray(encodedArray, GetMaxLengthOfBitsFromImage(encodedArray));  Console.WriteLine("Image after Huffman Decoder:");  **var** decodedArray = DecodeBinarySequence(encodedArray);  print2DArray(decodedArray);  Console.WriteLine("Image after Decompression:");  **var** decompressedArray = DecompressImage(i, decodedArray);  print2DArray(decompressedArray);  }  /// <summary>  /// Gets the longest element from the array and returns the length, used for formatting when printing to console.  /// </summary>  /// <param name="encodedArray"></param>  /// <returns></returns>  **private** **static** **int** **GetMaxLengthOfBitsFromImage**(**string**[,] encodedArray)  {  **int** longestBit = **0**;  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  **if** (encodedArray[i, j].Length > longestBit)  longestBit = encodedArray[i, j].Length;  }  }  **return** longestBit;  }  /// <summary>  /// Runs these calculations:  /// Compression ratio  /// Bits/pixel for the compressed image  /// RMS Error (Should be zero)  /// </summary>  /// <param name="coefficients"></param>  **private** **static** **void** **RunCalculations**(**int** i, **int**[,] originalImage, **int**[,] coefficients)  {  **var** encodedArray = EncodeBinarySequence(coefficients);  **var** decodedArray = DecodeBinarySequence(encodedArray);  **var** decompressedArray = DecompressImage(i, decodedArray);  //For Compression Ratio, the number of bits of the original image is always 8x16x16 = 2048  //Cr = No / Nc = 2048 / Nc  **var** compressedBits = GetCompressedBits(encodedArray);  **double** Cr = **2048.0** / compressedBits;  Console.WriteLine($"Compression Ratio = 2048 / {compressedBits} = {Cr.ToString("**0.**##")}");  **var** averageBitsOfCompressedImage = compressedBits / **256**;  Console.WriteLine($"Average Bits/Pixel = {compressedBits} / 256 = {averageBitsOfCompressedImage.ToString("**0.**##")}");  **double** RMSError = GetRMSError(originalImage, decompressedArray);  Console.WriteLine($"RMS Error = {RMSError}");  }  /// <summary>  /// Gets the RMS error from original image vs decompressed image. Exact formula can be found in report.  /// </summary>  /// <param name="originalImage"></param>  /// <param name="decompressedArray"></param>  /// <returns></returns>  **private** **static** **double** **GetRMSError**(**int**[,] originalImage, **int**[,] decompressedArray)  {  **double** rmsValue = **0**;  //Get Sum  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  rmsValue += Math.Pow(originalImage[i,j] - decompressedArray[i,j], **2**);  }  }  //Divide by 256, and get Square Root  rmsValue = Math.Sqrt(rmsValue / **256**);  **return** rmsValue;  }  /// <summary>  /// Gets total sum of the lenght of the compressed bits of encoded image.  /// </summary>  /// <param name="encodedArray"></param>  /// <returns></returns>  **private** **static** **double** **GetCompressedBits**(**string**[,] encodedArray)  {  **double** total = **0**;  **foreach** (**var** pixel **in** encodedArray)  {  total += pixel.Length;  }  **return** total;  }  /// <summary>  /// Encodes the 16x16 image into a binary sequence as described by the Huffman encoder.  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **string**[,] EncodeBinarySequence(**int**[,] array)  {  **var** tempArray = **new** **string**[**16**, **16**];  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  tempArray[i, j] = HuffmanTableEncoder(array[i, j]);  }  }  **return** tempArray;  }  /// <summary>  /// Decodes the 16x16 image into a coefficient sequence as described by the Huffman decoder.  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **static** **int**[,] DecodeBinarySequence(**string**[,] array)  {  **var** tempArray = **new** **int**[**16**, **16**];  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  tempArray[i, j] = HuffmanTableDecoder(array[i, j]);  }  }  **return** tempArray;  }  /// <summary>  /// Decompresses a 16x16 image array by using the coefficients and the first bit.  /// </summary>  /// <param name="k">Index of Formula to use for decoding</param>  /// <param name="decodedArray"></param>  /// <returns></returns>  **private** **static** **int**[,] DecompressImage(**int** k, **int**[,] decodedArray)  {  **var** tempArray = **new** **int**[**16**, **16**];  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  **if** (i == **0** && j == **0**)  {  tempArray[i, j] = decodedArray[i, j];  }  **else** **if** (j == **0**)  {  tempArray[i, j] = tempArray[i - **1**, j] + decodedArray[i, j];  }  **else**  {  tempArray[i, j] = RunDecompressor(i, j, k, tempArray, decodedArray);  }  }  }  **return** tempArray;  }  /// <summary>  /// Runs what decompression method will be used, we use 'k' as the index to choose which formula was originally used.  /// </summary>  /// <param name="i"></param>  /// <param name="j"></param>  /// <param name="k"></param>  /// <param name="tempArray"></param>  /// <param name="decodedArray"></param>  /// <returns></returns>  **private** **static** **int** **RunDecompressor**(**int** i, **int** j, **int** k, **int**[,] tempArray, **int**[,] decodedArray)  {  **if** (i == **0**)  {  **return** tempArray[i, j - **1**] + decodedArray[i, j];  }  **else** **if** (j == **0**)  {  **return** tempArray[i - **1**, j] + decodedArray[i - **1**, j];  }  **else**  {  **switch** (k)  {  **case** **1**:  **return** tempArray[i, j - **1**] + decodedArray[i, j];  **case** **2**:  **return** tempArray[i - **1**, j] + decodedArray[i, j];  **case** **3**:  **return** tempArray[i - **1**, j - **1**] + decodedArray[i, j];  **case** **4**:  **return** (tempArray[i, j - **1**] + tempArray[i - **1**, j] - tempArray[i - **1**, j - **1**]) + decodedArray[i, j];  **case** **5**:  **var** x = (**int**)Math.Floor((**double**)((tempArray[i - **1**, j] - tempArray[i - **1**, j - **1**]) / **2**));  **return** (tempArray[i, j - **1**] + x) + decodedArray[i, j];  **case** **6**:  **var** y = (**int**)Math.Floor((**double**)((tempArray[i, j - **1**] - tempArray[i - **1**, j - **1**]) / **2**));  **return** (tempArray[i - **1**, j] + y) + decodedArray[i, j];  **case** **7**:  **var** z = (**int**)Math.Floor((**double**)((tempArray[i, j - **1**] + tempArray[i - **1**, j]) / **2**));  **return** z + decodedArray[i, j];  **default**:  **return** -**1**;  }  }  }  /// <summary>  /// Initializes an array containing the coefficients for the encoded image. First Row is always X = A, first column is always X = B.  /// </summary>  /// <param name="array"></param>  /// <returns></returns>  **private** **static** **int**[,] InitializeCoefficientsArray(**int**[,] array)  {  **int**[,] tempArray = **new** **int**[**16**, **16**];  **if** (array.Length == **256**)  {  //Initialize first element  tempArray[**0**, **0**] = array[**0**, **0**];  //First Row  **for** (**int** i = **1**; i < **16**; i++)  {  tempArray[**0**, i] = DoXEqualsA(array, **0**, i);  }  //First Column  **for** (**int** j = **1**; j < **16**; j++)  {  tempArray[j, **0**] = DoXEqualsB(array, j, **0**);  }  }  **return** tempArray;  }  /// <summary>  /// Method to print out a 16 x 16 string array to the console neatly.  /// </summary>  /// <param name="array"></param>  **static** **void** **print2DArray**(**int**[,] array)  {  Console.Write(Environment.NewLine);  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  Console.Write(array[i, j].ToString().PadLeft(**4**));  }  Console.Write(Environment.NewLine);  Console.Write(Environment.NewLine);  }  }  /// <summary>  /// Method to print out a 16 x 16 int array to the console neatly.  /// </summary>  /// <param name="array"></param>  /// <param name="pad"></param>  **static** **void** **print2DArray**(**string**[,] array, **int** pad = **4**)  {  Console.Write(Environment.NewLine);  **for** (**int** i = **0**; i < **16**; i++)  {  **for** (**int** j = **0**; j < **16**; j++)  {  Console.Write(array[i, j].PadLeft(pad));  }  Console.Write(Environment.NewLine);  Console.Write(Environment.NewLine);  }  }  }  } |