Example of how information may appear

* Description: ??
  + Size = ??
  + Location = ??

GIF files consist of a header and different sections with a one-byte identifier in front.

Identifiers are as follows:

0x21 -> Extension Block

0x2C -> Image Descriptor

0x3B -> End of data. Should be the last byte.

The other identifiers specify what type of extension block you have

Those are as follows: (Note that 0x3B is not included in these)

0x00-0x7F -> Graphic Rendering Blocks

0x80-0xF9 -> Control Blocks

0xFA-0xFF -> Special Purpose Blocks

When decompressing, you need to keep your dictionary going until you are instructed to clear it or you hit the end of data symbol. This means that you need to collect all of the data in a sub image and decompress it all together, or hard code a more customized version into your code.

For the header:

* Description: GIF version used
  + Size = 6
  + Location = 0
* Description: Width
  + Size = 2
  + Location = 6
* Description: Height
  + Size = 2
  + Location = 8
* Description: Packed Fields
  + Size = 1
  + Location = 9
    - Global table flag – 1 bit
    - Color res – 3 bits (add 1)
    - Sort flag – 1 bit (whether the table is sorted by importance)
    - Size of global table – 3 bits
* Description: Background color index
  + Size = 1
  + Location = 10
* Description: Pixel Aspect Ratio
  + Size = 1
  + Location = 11
* Description: Global Color Table (May not exist)
  + Size = 2^(SizeOfGlobalTable+1)
  + Location = 12

For the Image Descriptor:

(Note that the location of the data is now unknown and won’t be mentioned)

(The data will be in the order that they will be read though)

* Description: ID (Must be 0x2C)
  + Size = 1
* Description: Left Pos
  + Size = 2
* Description: Top Pos
  + Size = 2
* Description: Width
  + Size = 2
* Description: Height
  + Size = 2
* Description: Packed Field
  + Size = 1
    - Local color table flag – 1 bit
    - Interlace Flag – 1 bit
    - Sort Flag – 1 bit
    - Reserved – 2 bits
    - Size of Local Color Table – 3 bits
* Description: Local Color Table
  + Size = 2^(SizeOfLocalColorTable + 1)
* Description: LZW Minimum Code Size
  + Size = 1
* Description: Image Data (In Sub Blocks)
  + Size = ??

For the Graphic Control Extension:

* Description: ID (0x21)
  + Size = 1
* Description: Label (0xF9)
  + Size = 1
* Description: Block size
  + Size = 1
* Description: Packed Field
  + Size = 1
    - Reserved – 3 bits
    - Disposal Method – 3 bits
    - User Input Flag – 1 bit
    - Transparent Color Flag – 1 bit
* Description: Delay Time ( 1/100 secs)
  + Size = 2
* Description: Transparent Color Index
  + Size = 1
* Description: Block Terminator
  + Size = 1

For Comment Blocks: (We will skip these)

* Description: ID (Must be 0x21)
  + Size = 1
* Description: Label (Must be 0xFE)
  + Size = 1
* Description: Data Sub Blocks (Comment Data)
  + Size = ??
    - Size in bytes comes first then the data.
    - Ending with a Block Terminator. (0x00)
* Description: Block Terminator (end of the Comment Extension block)
  + 0x00
  + Size = 1

For Plain Test Blocks: (We will skip these)

* Description: ID (Must be 0x21)
  + Size = 1
* Description: Label (Must be 0xFF)
  + Size = 1
* Description: Data Size
  + Size = 1
* Description: More Stuff (Skipping because it is a lot and unnecessary)
  + Size = ??
* Description: Block Terminator (end of the Comment Extension block)
  + (Separate from the sub blocks)
  + Size = 1

For Sub Blocks:

* Description: Block Size
  + Size = 1
* Description: Data
  + Size = Block Size
    - Always read as bytes
* Description: Block Terminator (0x00)
  + (Technically, this doesn’t exist and is just a sub block with size 0)
  + (It means that there is no more sub blocks)
  + Size = 1

At the end of your file, 0x3B will appear.

For dealing with Interlaced Images,

The rows of an Interlaced images are arranged in the following order:

Group 1 : Every 8th. row, starting with row 0. (Pass 1)

Group 2 : Every 8th. row, starting with row 4. (Pass 2)

Group 3 : Every 4th. row, starting with row 2. (Pass 3)

Group 4 : Every 2nd. row, starting with row 1. (Pass 4)