# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | |
|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

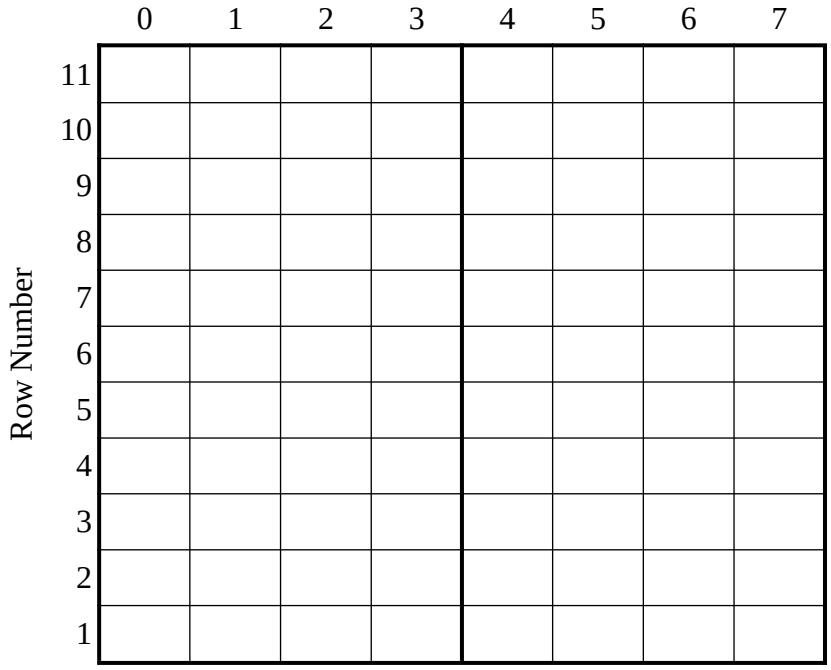
# Worksheet: Reading a Bitmap File

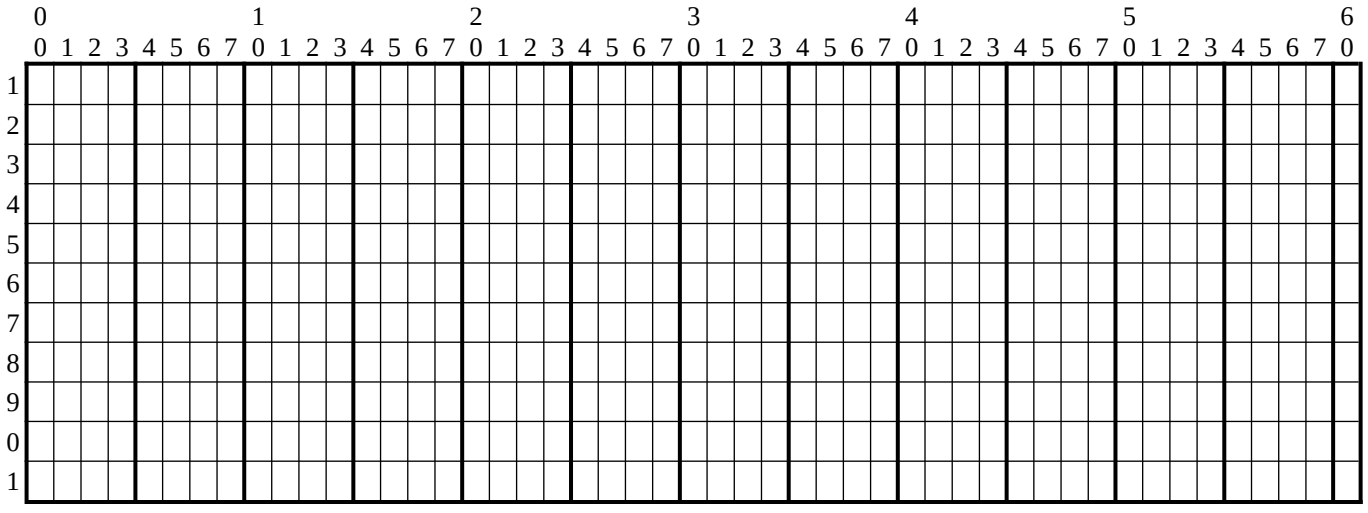


`Angel.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | 80 | 00 | FF | FF |
| 0000090 | 00 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | F1 | FF | FF | FF |
| 00000A0 | 80 | 00 | FF | FF | EE | FF | FF | FF | 80 | 00 | FE | EB | BE | C7 | 1F | F5 |
| 00000B0 | 80 | 00 | FE | EB | B0 | BF | BF | F5 | 80 | 00 | FE | 0B | AE | 83 | BF | E0 |
| 00000C0 | 80 | 00 | FE | E9 | AE | BB | BF | F5 | 80 | 00 | BA | EA | 70 | C7 | BB | A0 |
| 00000D0 | 80 | 00 | D7 | 5F | FF | FF | BD | 75 | 80 | 00 | EF | BF | FF | FF | 3E | F5 |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

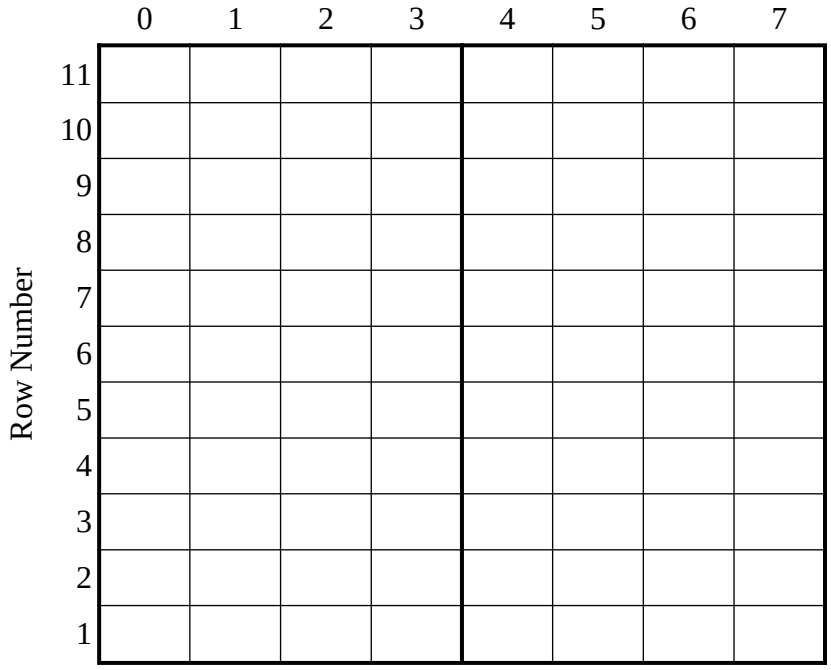
# Worksheet: Reading a Bitmap File

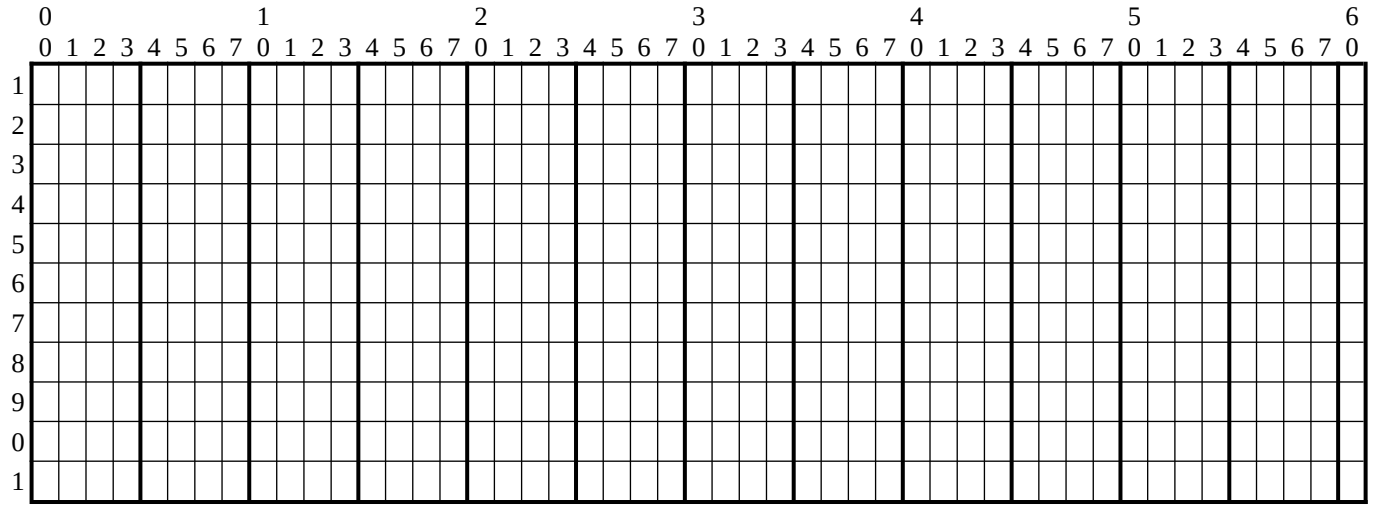


`Arthur.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 2B | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | 00 | FF |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | E0 | 00 | 00 | FF | FF | FF | FF | FF | E0 |
| 00000A0 | 00 | 00 | DD | 7F | 37 | 65 | 7F | 60 | 00 | 00 | DD | 7E | D7 | 59 | 7F | E0 |
| 00000B0 | 00 | 00 | C1 | 7E | F7 | 5D | 7F | 60 | 00 | 00 | DD | 36 | F3 | 5D | 37 | 60 |
| 00000C0 | 00 | 00 | DD | 4C | 34 | DD | 4F | 60 | 00 | 00 | EB | FE | F7 | FF | FF | 60 |
| 00000D0 | 00 | 00 | F7 | FE | F7 | FF | FF | 60 | 00 | 00 | FF | FF | FF | FF | FF | E0 |
| 00000E0 | 00 | 00 | FF | FF | FF | FF | FF | E0 | 00 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

# Worksheet: Reading a Bitmap File



`Bettina.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | 80 | 00 | CB | C0 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000A0 | 80 | 00 | BE | 1C | 79 | E7 | 1B | B0 | 80 | 00 | BF | 6B | F6 | DB | BB | AE |
| 00000B0 | 80 | 00 | DF | 68 | 37 | DF | BB | B0 | 80 | 00 | EF | 1B | B7 | DF | B9 | BE |
| 00000C0 | 80 | 00 | F7 | 6C | 61 | 87 | 3A | 71 | 80 | 00 | FB | 6F | F7 | DF | FF | FF |
| 00000D0 | 80 | 00 | FA | 1F | F7 | DF | BF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number

| Row Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 11 | | | | | | | | |
| 10 | | | | | | | | |
| 9 | | | | | | | | |
| 8 | | | | | | | | |
| 7 | | | | | | | | |
| 6 | | | | | | | | |
| 5 | | | | | | | | |
| 4 | | | | | | | | |
| 3 | | | | | | | | |
| 2 | | | | | | | | |
| 1 | | | | | | | | |

| | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | | 4 | | | | | | | | 5 | | | | | | | | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

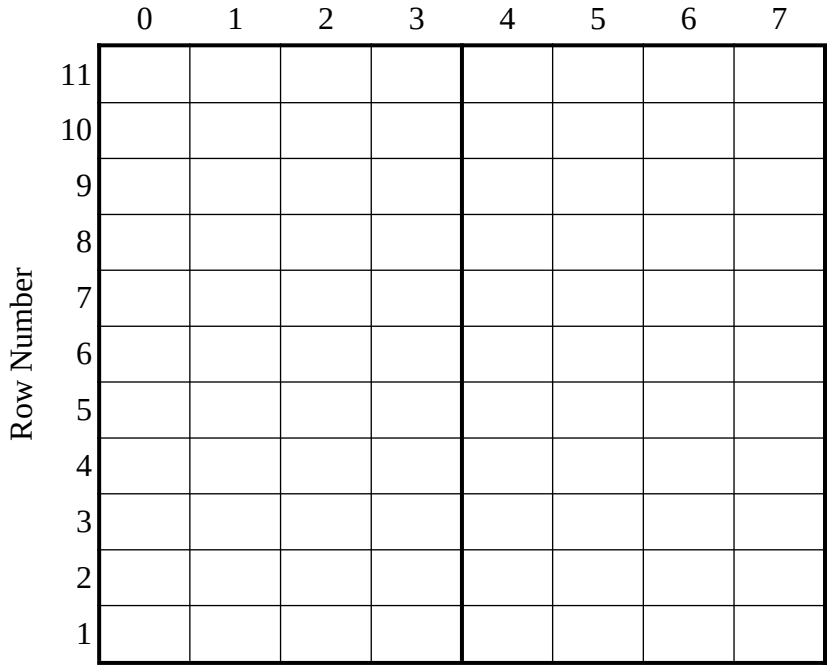
# Worksheet: Reading a Bitmap File

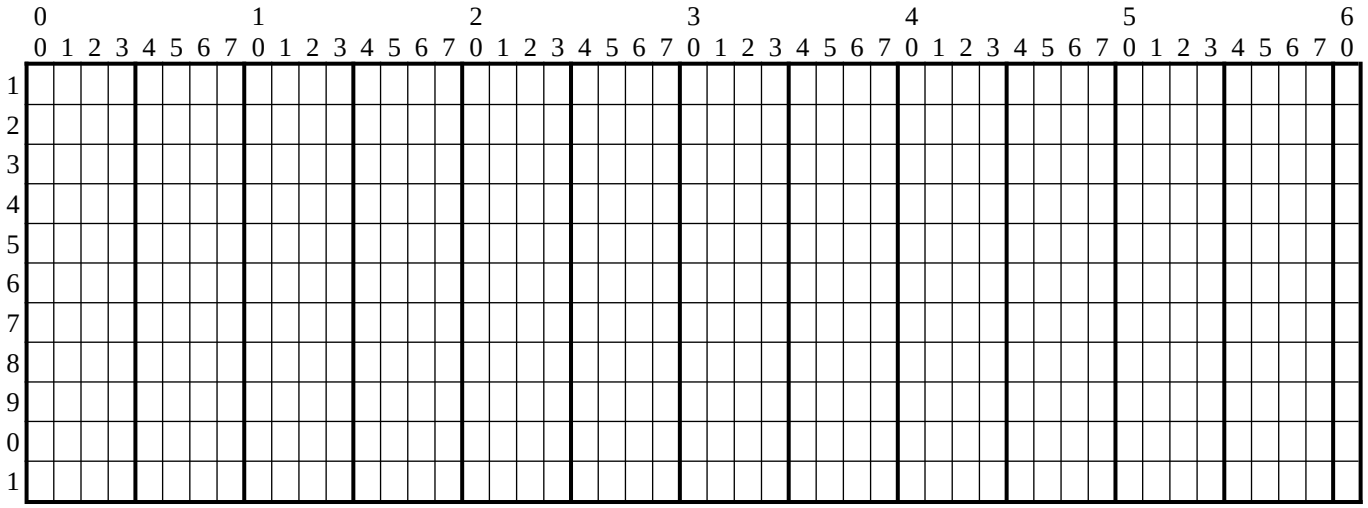


Buddy.bmp

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | 00 | 00 | D7 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | 1F | FF |
| 00000A0 | 80 | 00 | FF | FF | FF | FE | EF | FF | 80 | 00 | FE | 1C | B2 | CB | EF | F7 |
| 00000B0 | 80 | 00 | FF | 6B | 2C | B3 | 2F | FB | 80 | 00 | FF | 6B | AE | BA | CF | FD |
| 00000C0 | 80 | 00 | 83 | 1B | AC | B2 | E8 | 3E | 80 | 00 | FF | 6B | B2 | CA | EF | FD |
| 00000D0 | 80 | 00 | FF | 6F | FE | FB | FF | FB | 80 | 00 | FE | 1F | FE | FB | FF | F7 |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | |
|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

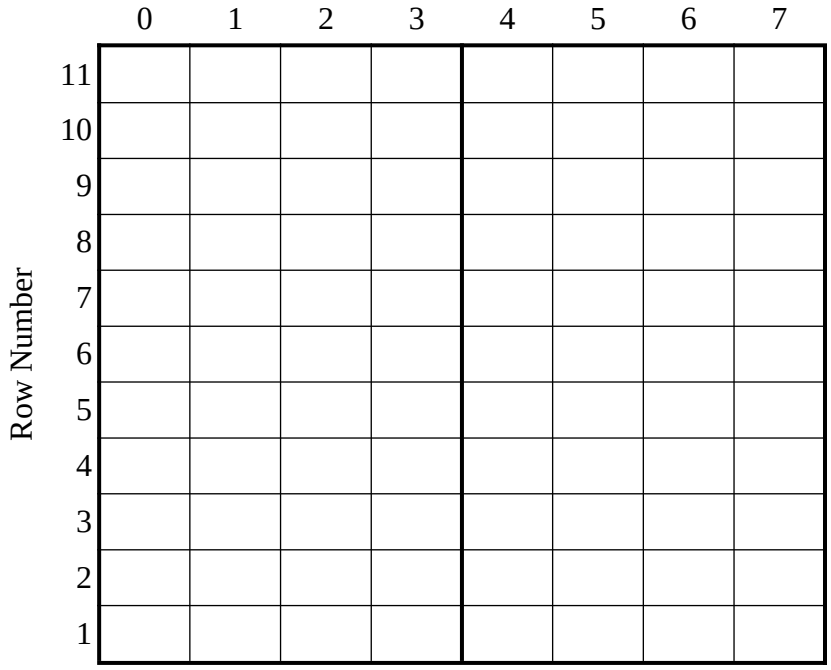
# Worksheet: Reading a Bitmap File

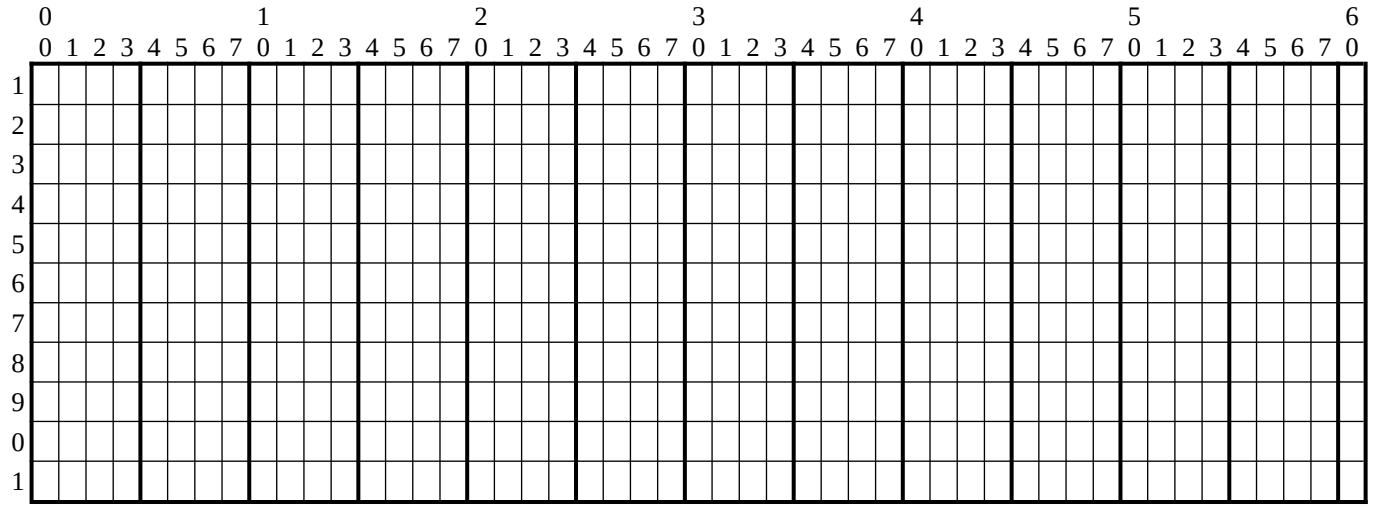


`David.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | C0 | C0 | C0 | 00 | 00 | 00 |
| 0000090 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000A0 | 80 | 00 | F7 | F0 | E1 | DE | 39 | 7F | 80 | 00 | EF | FB | 5D | AF | 76 | 7F |
| 00000B0 | 80 | 00 | DF | FB | 61 | AF | 77 | 7F | 80 | 00 | BC | 1B | 7D | 77 | 76 | 41 |
| 00000C0 | 80 | 00 | DF | FB | 63 | 76 | 79 | 7F | 80 | 00 | EF | FB | 7F | FF | FF | 7F |
| 00000D0 | 80 | 00 | F7 | F0 | FF | FF | 7F | 7F | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

2. a) Write the first two bytes of the file in hexadecimal format.

   b0 Convert these two bytes to their ASCII equivalent.

3. a) Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file.

4. a) Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian).

5. a) Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal.

   b) Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124).

6. a) Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary.

| | hexadecimal | denary |
|---|---|---|
| width | | |
| height | | |

7. a) Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary.

8. a) Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. 0

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

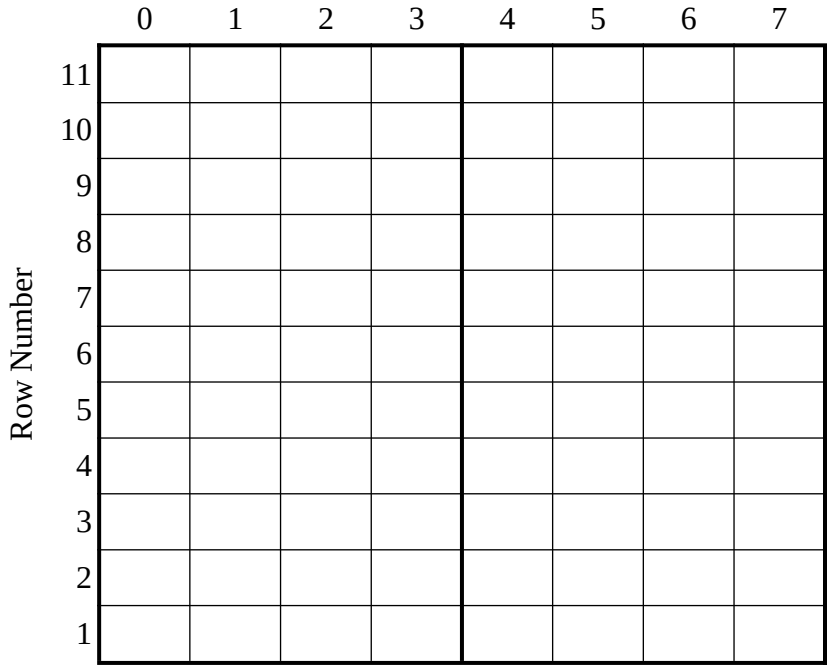
# Worksheet: Reading a Bitmap File

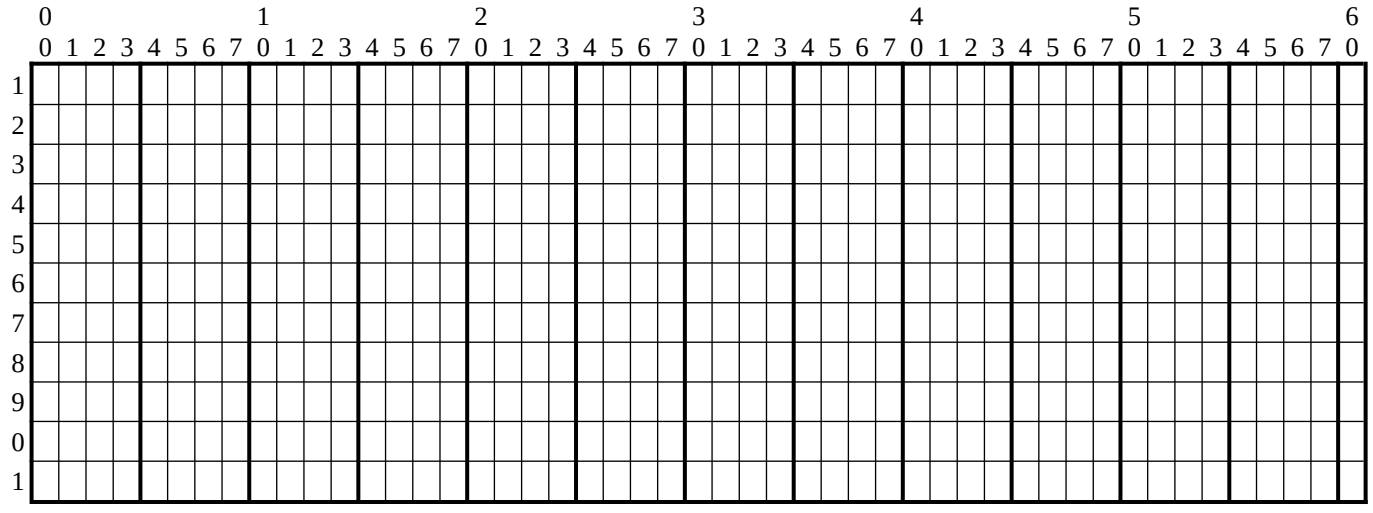


`Eline.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | CB | C0 | FF | 00 | 80 | 00 |
| 0000090 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000A0 | 80 | 00 | FE | 0C | 71 | BB | 1F | FF | 80 | 00 | FE | FE | FB | BA | FF | FF |
| 00000B0 | 80 | 00 | 82 | FE | FB | BA | 0F | E0 | 80 | 00 | FE | 1E | FB | 9A | E8 | 3F |
| 00000C0 | 80 | 00 | 82 | FE | F3 | A7 | 1F | E0 | 80 | 00 | FE | FE | FF | FF | FF | FF |
| 00000D0 | 80 | 00 | FE | 0C | FB | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

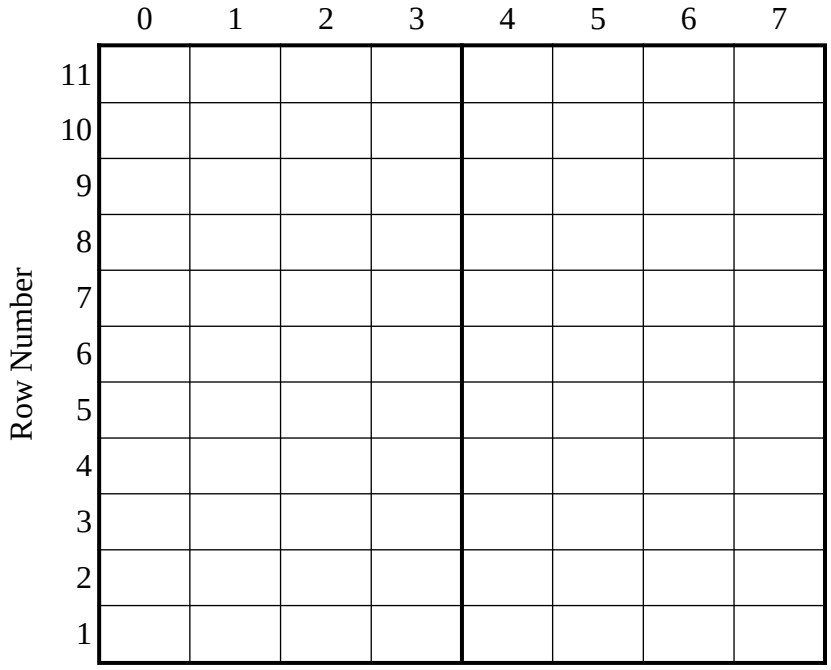
# Worksheet: Reading a Bitmap File

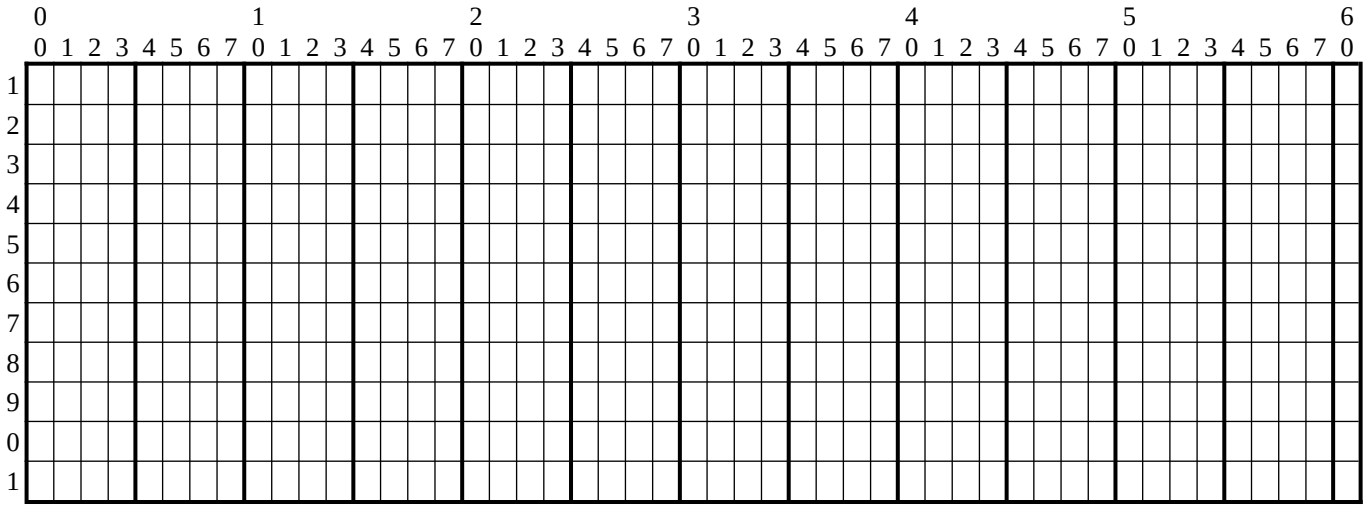


`Gavin.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 80 | 80 | 00 | 00 | A5 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | BF | FF | FF | FF | FB |
| 00000A0 | 80 | 00 | EF | 1C | 70 | EF | 1B | B1 | 80 | 00 | C7 | BB | AE | D7 | BB | BB |
| 00000B0 | 80 | 00 | EB | FB | 30 | D7 | BB | BF | 80 | 00 | C7 | FB | FE | BB | B9 | BF |
| 00000C0 | 80 | 00 | AF | FB | F1 | BB | 3A | 7F | 80 | 00 | C7 | FB | BF | FF | FF | FF |
| 00000D0 | 80 | 00 | EF | FC | 7F | FF | BF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

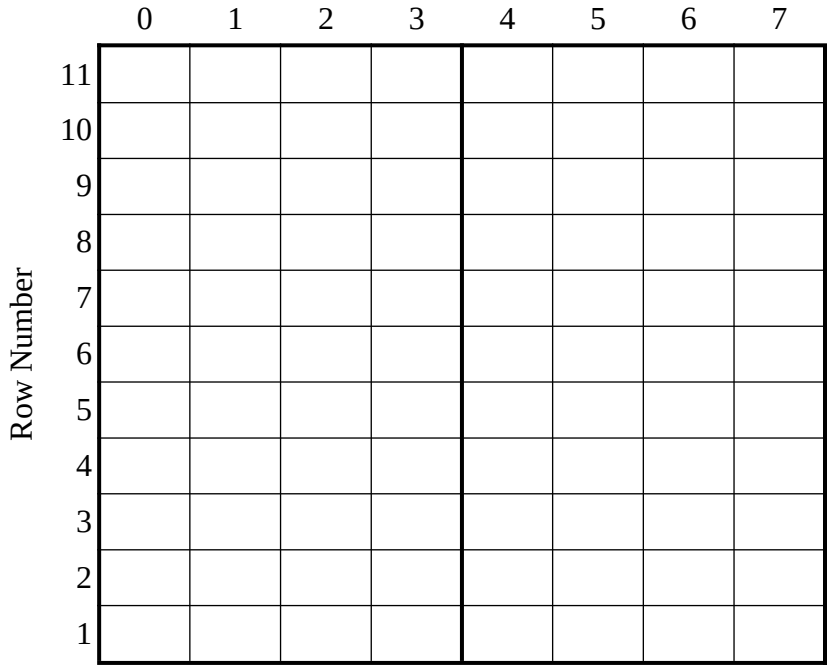
# Worksheet: Reading a Bitmap File

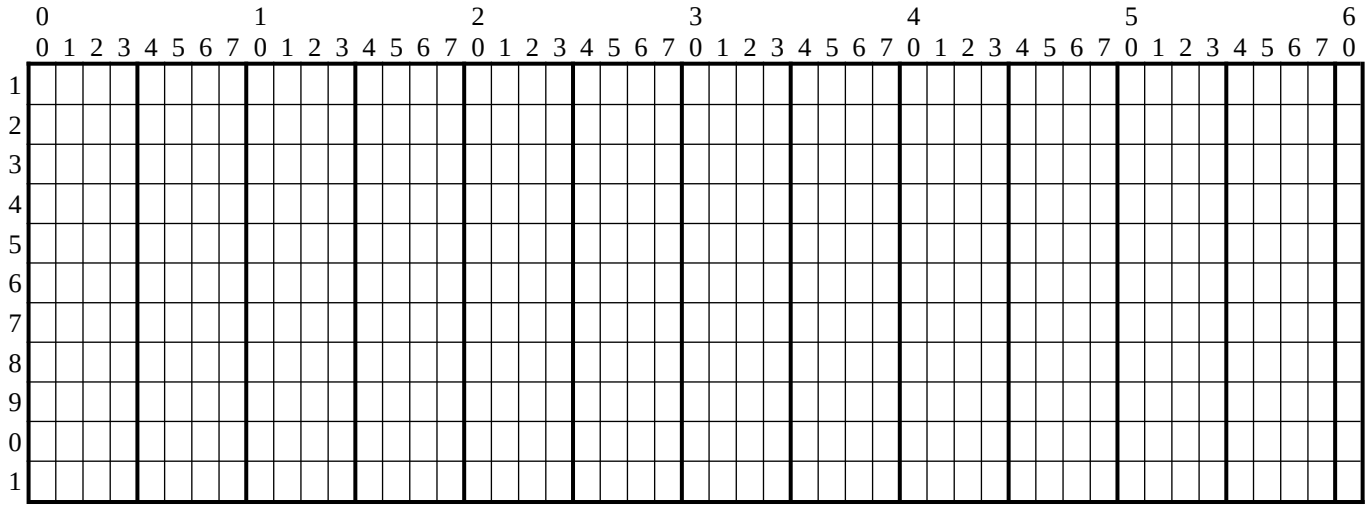


`Harry.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | 00 | 00 | 00 | 80 |
| 0000090 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FC | 7F |
| 00000A0 | 80 | 00 | FE | 0F | FF | FF | FB | A0 | 80 | 00 | FF | FB | B0 | BE | FF | BF |
| 00000B0 | 80 | 00 | BB | FB | AE | BE | FC | BF | 80 | 00 | D7 | FB | B0 | BE | FB | 3F |
| 00000C0 | 80 | 00 | 83 | F8 | 3E | 9A | 6B | BF | 80 | 00 | D7 | FB | B1 | A6 | 9B | BF |
| 00000D0 | 80 | 00 | BB | FB | BF | FF | FF | FF | 80 | 00 | FF | FB | BF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

2. a) Write the first two bytes of the file in hexadecimal format.

   b0 Convert these two bytes to their ASCII equivalent.

3. a) Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file.

4. a) Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian).

5. a) Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal.

   b) Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124).

6. a) Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary.

| | hexadecimal | denary |
|---|---|---|
| width | | |
| height | | |

7. a) Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary.

8. a) Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number.

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

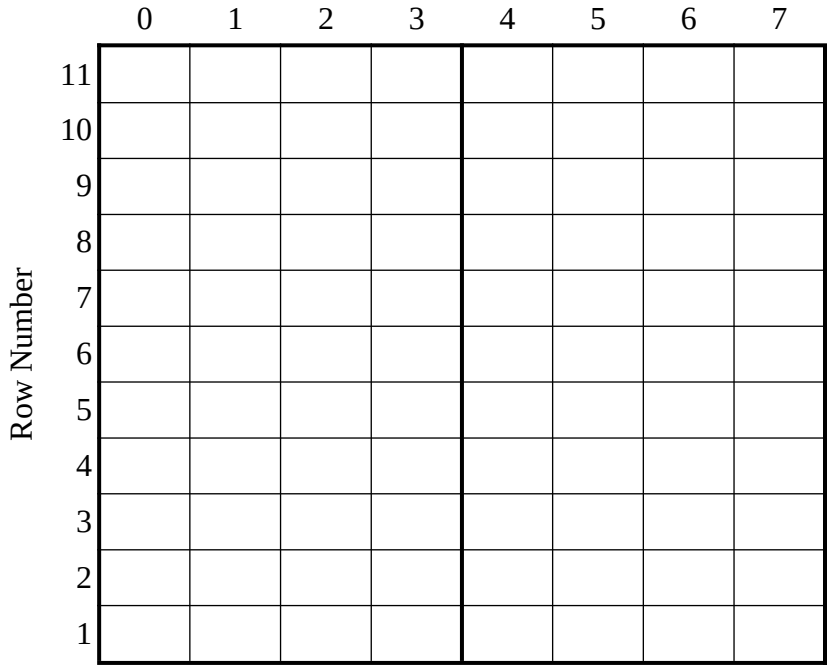
# Worksheet: Reading a Bitmap File

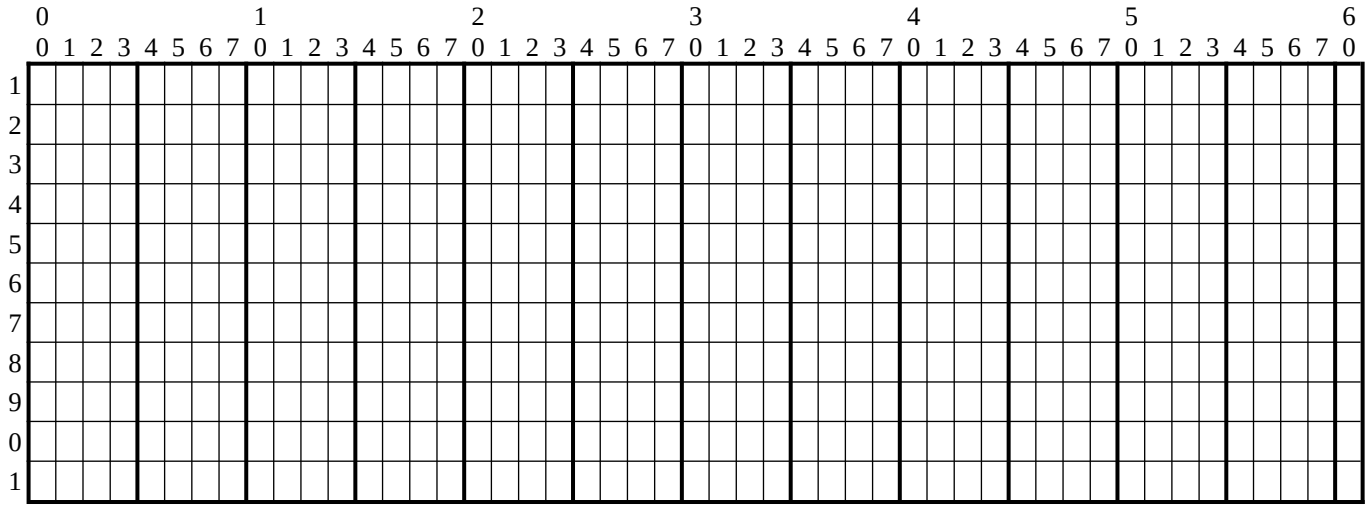


`Ivy.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 80 | 00 | 00 | FF |
| 0000090 | 00 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | 1F | FF |
| 00000A0 | 80 | 00 | FF | FF | FF | FE | EF | FF | 80 | 00 | FF | FF | 71 | EF | ED | FF |
| 00000B0 | 80 | 00 | FF | FE | FB | D7 | 2E | FF | 80 | 00 | FE | 0D | FB | D6 | CF | 7F |
| 00000C0 | 80 | 00 | FF | FD | FB | BA | EF | 7F | 80 | 00 | B6 | 0D | FB | BA | EF | 6D |
| 00000D0 | 80 | 00 | AB | FE | FB | FF | FE | EA | 80 | 00 | DB | FF | 71 | FF | FD | F6 |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | |
|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

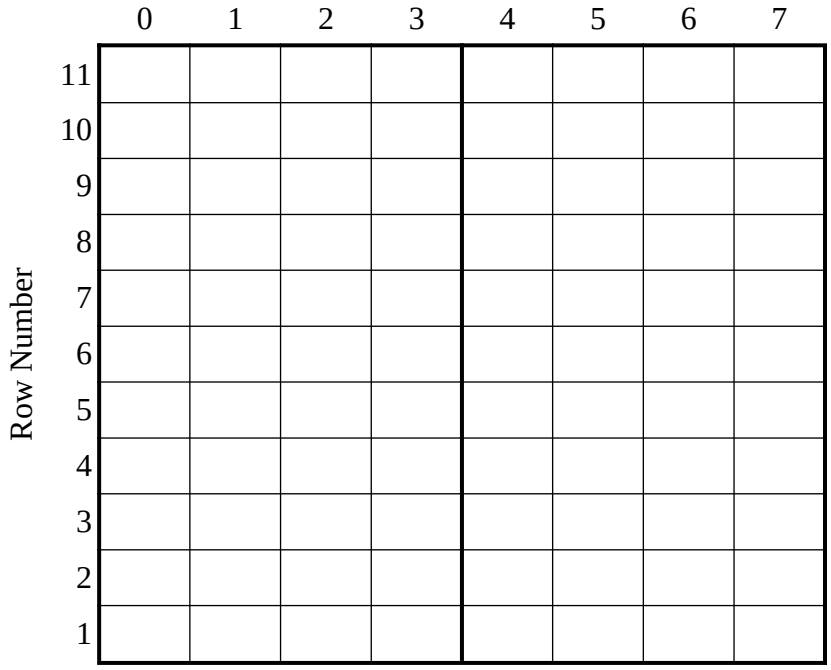
# Worksheet: Reading a Bitmap File

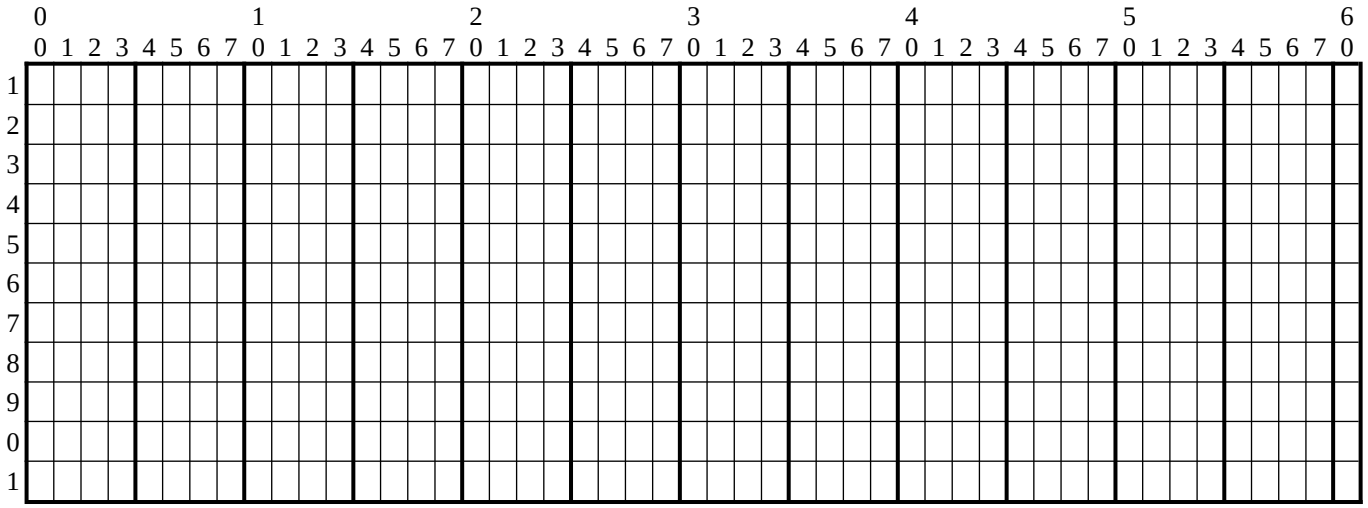


`Joey.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 80 | 00 | 00 | 80 | 00 |
| 0000090 | 00 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | 1F | FF |
| 00000A0 | 80 | 00 | FF | FF | FF | FE | EF | FF | 80 | 00 | F7 | FC | F1 | C7 | EF | F7 |
| 00000B0 | 80 | 00 | EF | FB | 6E | BF | 2F | FB | 80 | 00 | DE | 0F | 6E | 82 | C8 | 3D |
| 00000C0 | 80 | 00 | DF | FF | 6E | BA | EF | FD | 80 | 00 | DE | 0F | 71 | C6 | E8 | 3D |
| 00000D0 | 80 | 00 | EF | FF | 7F | FF | FF | FB | 80 | 00 | F7 | FE | 3F | FF | FF | F7 |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

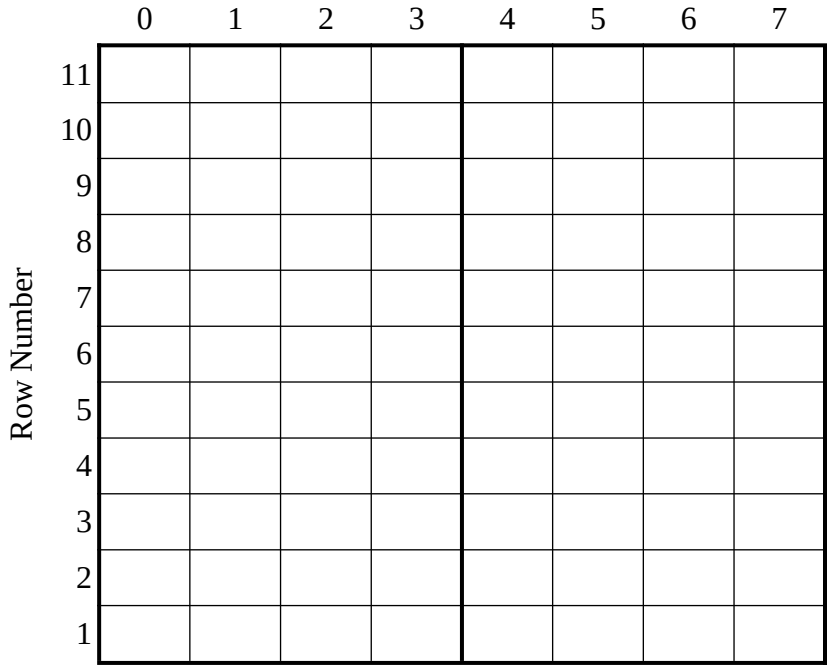
# Worksheet: Reading a Bitmap File

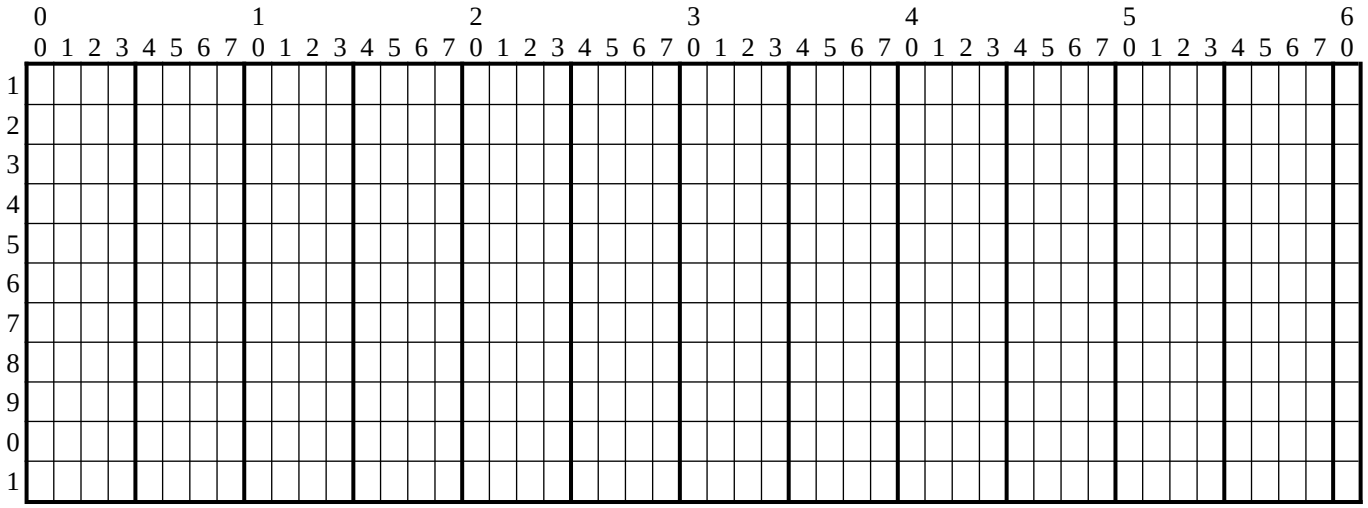


Laura.bmp

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | FF | 00 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | 83 | F8 | 3F | FF | FF | FF |
| 00000A0 | 80 | 00 | FF | BF | E0 | C3 | 2B | F0 | 80 | 00 | FF | BF | EF | BA | CB | EE |
| 00000B0 | 80 | 00 | FF | BF | EF | C2 | EB | F0 | 80 | 00 | FF | BF | EF | FA | E9 | BE |
| 00000C0 | 80 | 00 | FF | BF | EF | C6 | EA | 71 | 80 | 00 | FF | BF | EF | FF | FF | FF |
| 00000D0 | 80 | 00 | FF | BF | EF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | |
|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | |
| | b0 | Convert these two bytes to their ASCII equivalent. | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | |
|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

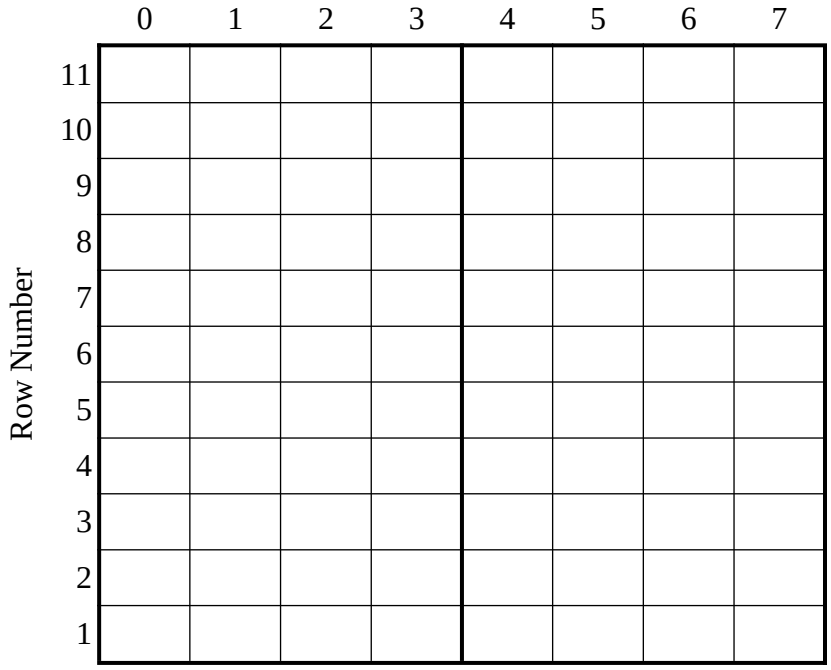
# Worksheet: Reading a Bitmap File

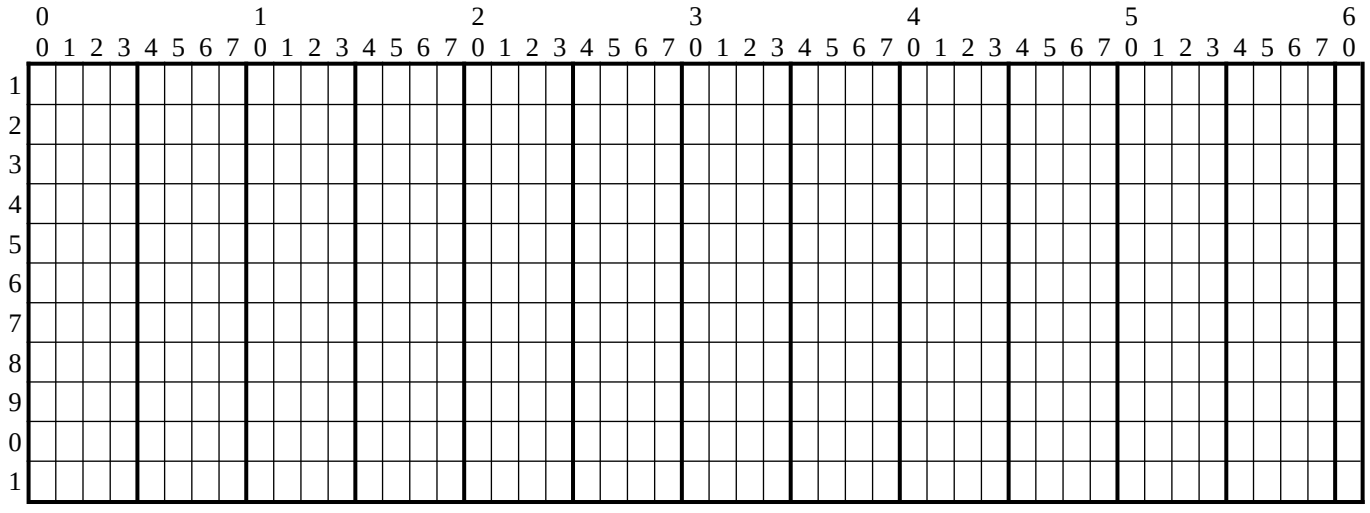


Owen.bmp

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 2A | 2A | A5 | 00 | 00 | A5 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000A0 | 80 | 00 | FF | DC | 75 | C6 | ED | FF | 80 | 00 | FF | BB | AA | BE | EE | FF |
| 00000B0 | 80 | 00 | FF | 7B | AA | 82 | EF | 7F | 80 | 00 | 83 | 7B | AE | BA | 6F | 60 |
| 00000C0 | 80 | 00 | FF | 7B | AE | C6 | 9F | 7F | 80 | 00 | FF | BB | BF | FF | FE | FF |
| 00000D0 | 80 | 00 | FF | DC | 7F | FF | FD | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

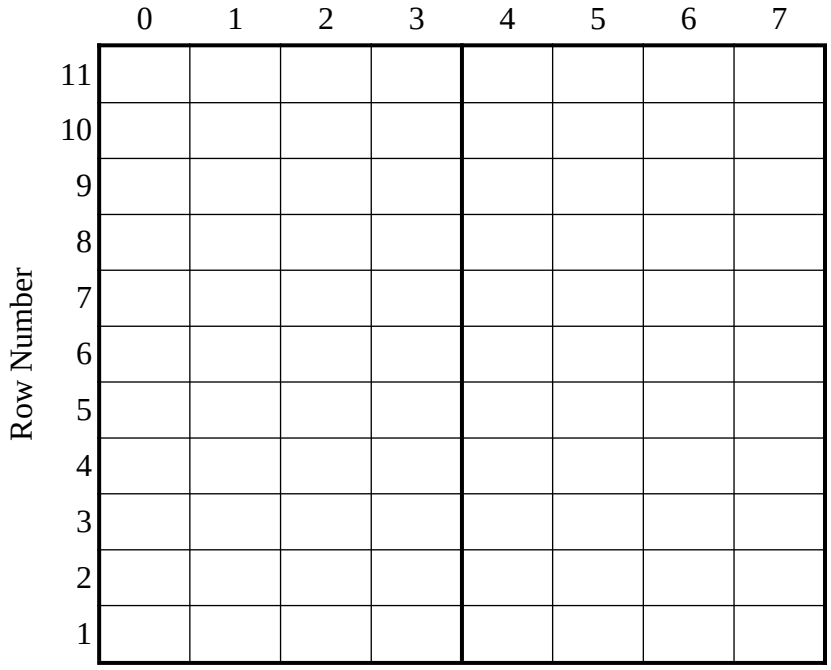
# Worksheet: Reading a Bitmap File

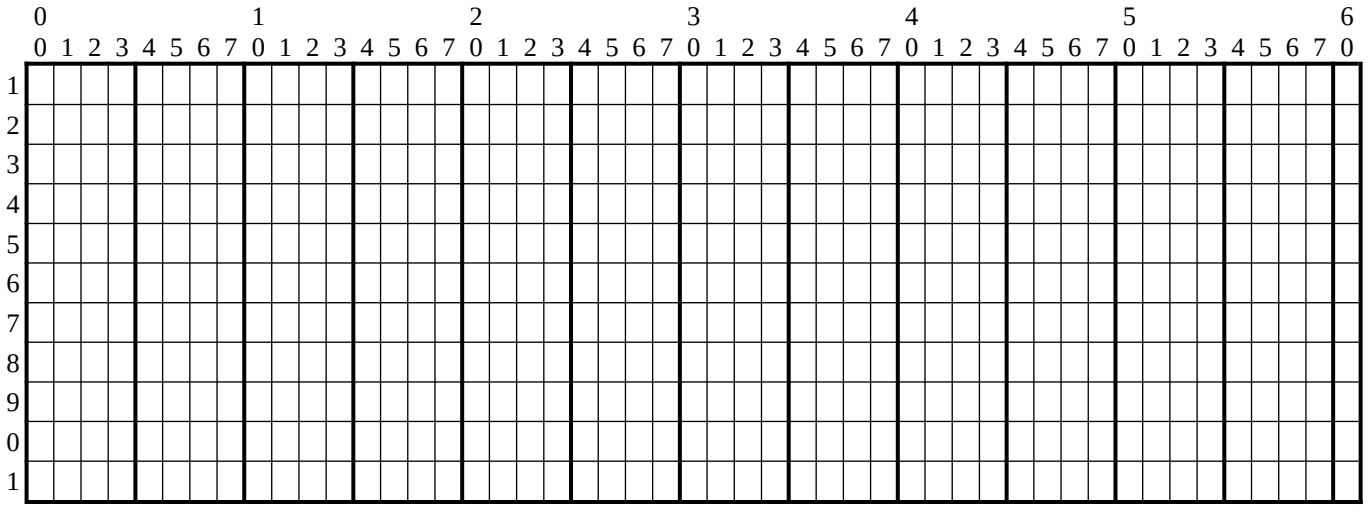


`Rocky.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | C0 | C0 | C0 | 00 | 2A | 2A |
| 0000090 | A5 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | E3 | FF | FF |
| 00000A0 | 80 | 00 | FF | FF | FF | DD | FF | FF | 80 | 00 | DD | 8E | 37 | 7D | DF | FD |
| 00000B0 | 80 | 00 | DB | 75 | D6 | E5 | DF | FD | 80 | 00 | D7 | 75 | F1 | D9 | DC | 1D |
| 00000C0 | 80 | 00 | C3 | 75 | D6 | DD | DF | FD | 80 | 00 | DD | 8E | 37 | 5D | DC | 1D |
| 00000D0 | 80 | 00 | DD | FF | F7 | FF | DF | FD | 80 | 00 | C3 | FF | F7 | FF | DF | FD |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

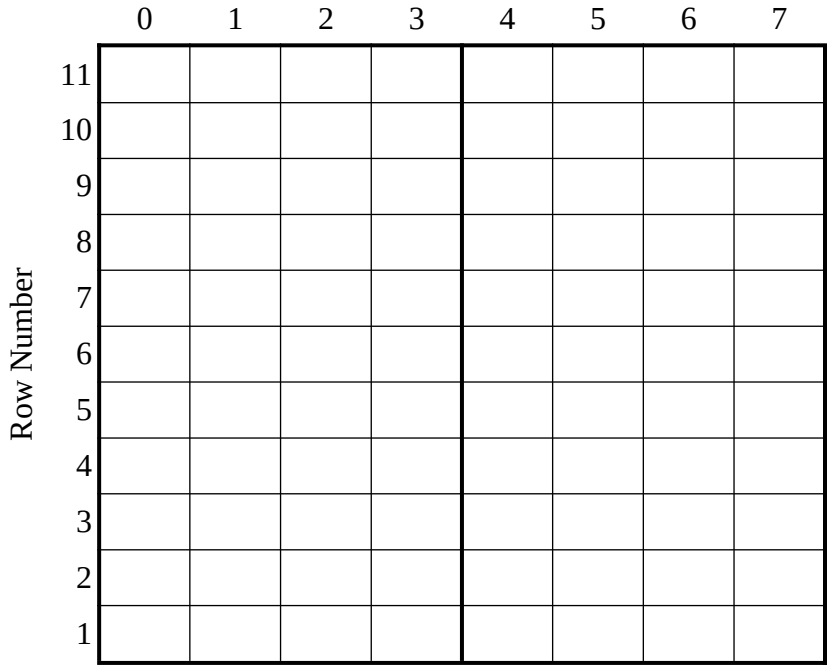
# Worksheet: Reading a Bitmap File

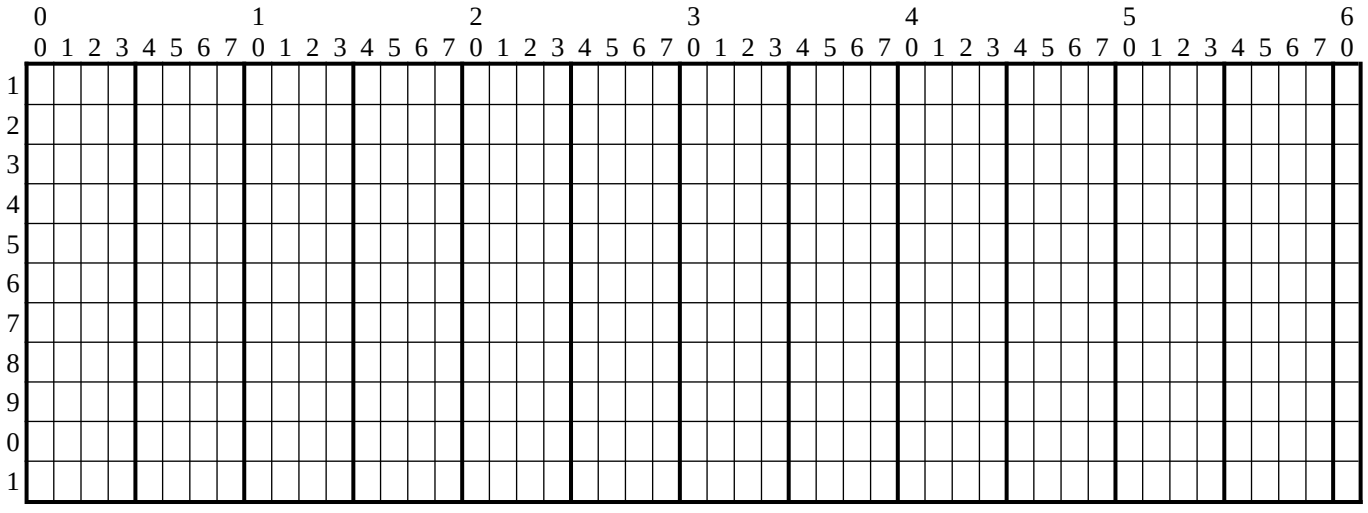


`Ruby.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | 80 | 00 | FF | 00 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | F1 | FF | FF | FF |
| 00000A0 | 80 | 00 | FF | FF | EE | FF | FF | FF | 80 | 00 | BB | 2A | 7E | FE | 0C | 72 |
| 00000B0 | 80 | 00 | B6 | C9 | B2 | FE | FE | EC | 80 | 00 | AE | EB | AC | FE | FE | EE |
| 00000C0 | 80 | 00 | 86 | E9 | AE | 82 | FE | EE | 80 | 00 | BA | EA | 6E | FE | FC | EE |
| 00000D0 | 80 | 00 | BB | FB | FF | FE | FF | FF | 80 | 00 | 87 | FB | FF | FE | FE | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

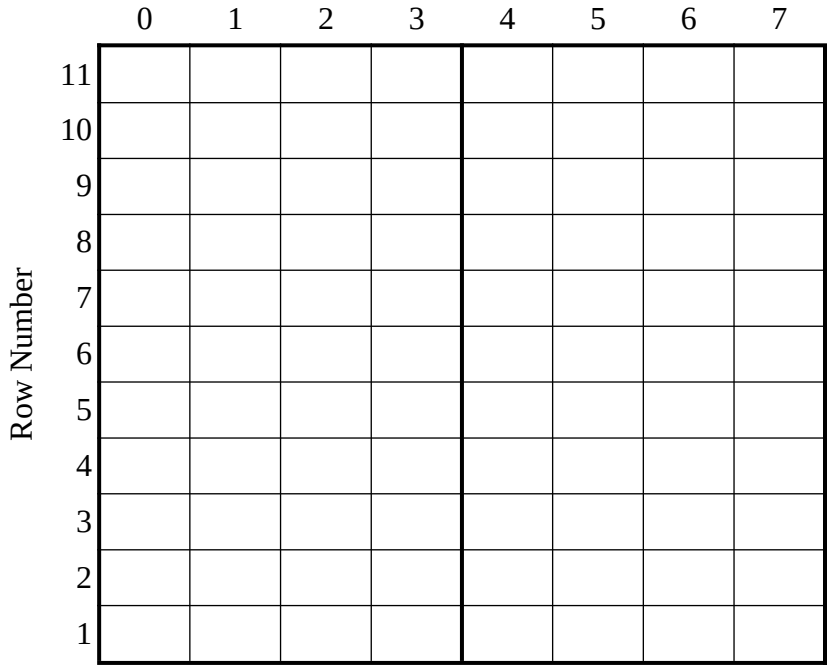
# Worksheet: Reading a Bitmap File

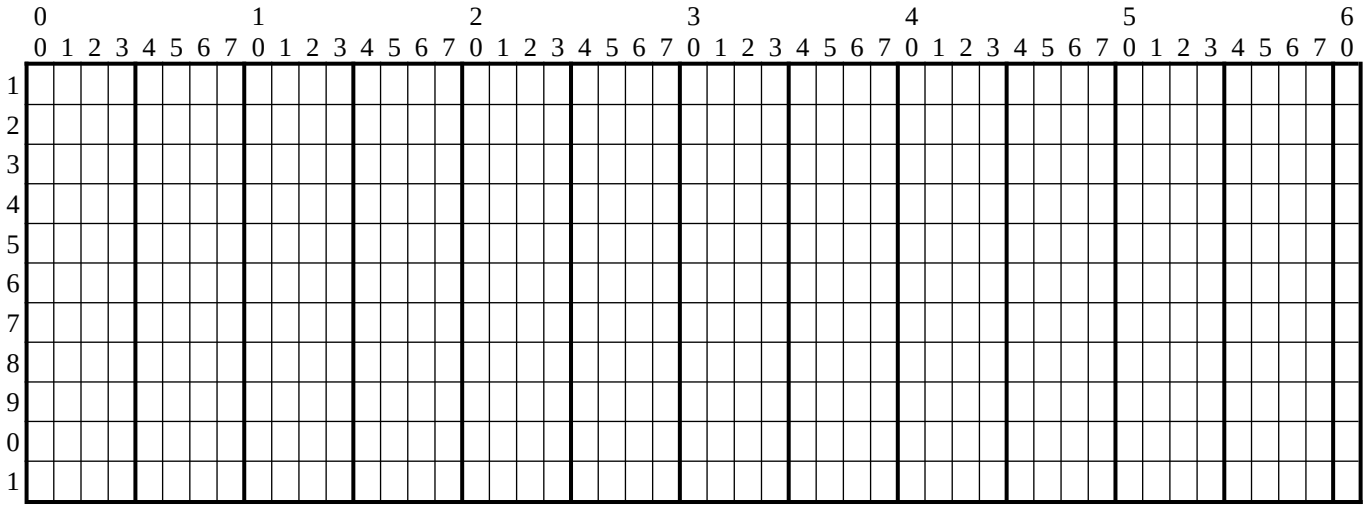


`Sarah.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | 00 | FF | 00 | 00 | 00 |
| 0000090 | FF | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | BF | FF |
| 00000A0 | 80 | 00 | C7 | 0B | F0 | BB | 1B | B2 | 80 | 00 | BA | EB | EE | BB | B9 | 2C |
| 00000B0 | 80 | 00 | FB | 0B | F0 | BB | FA | AE | 80 | 00 | C7 | E9 | BE | 9B | FA | AE |
| 00000C0 | 80 | 00 | BF | 1A | 71 | A7 | FB | AE | 80 | 00 | BB | FF | FF | BF | FB | BF |
| 00000D0 | 80 | 00 | C7 | FF | FF | BF | FB | BF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | |
|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | |
| | b0 | Convert these two bytes to their ASCII equivalent. | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | |
|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

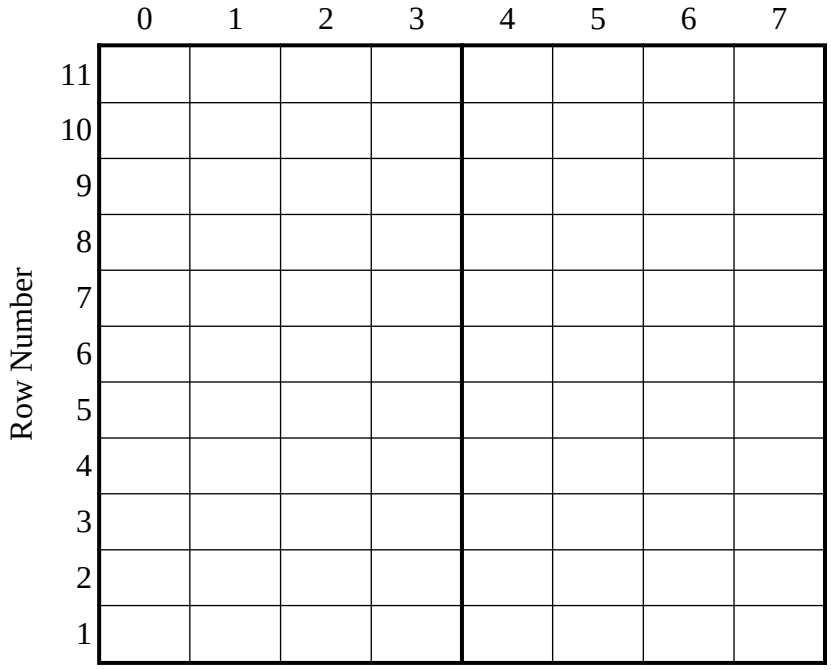
# Worksheet: Reading a Bitmap File

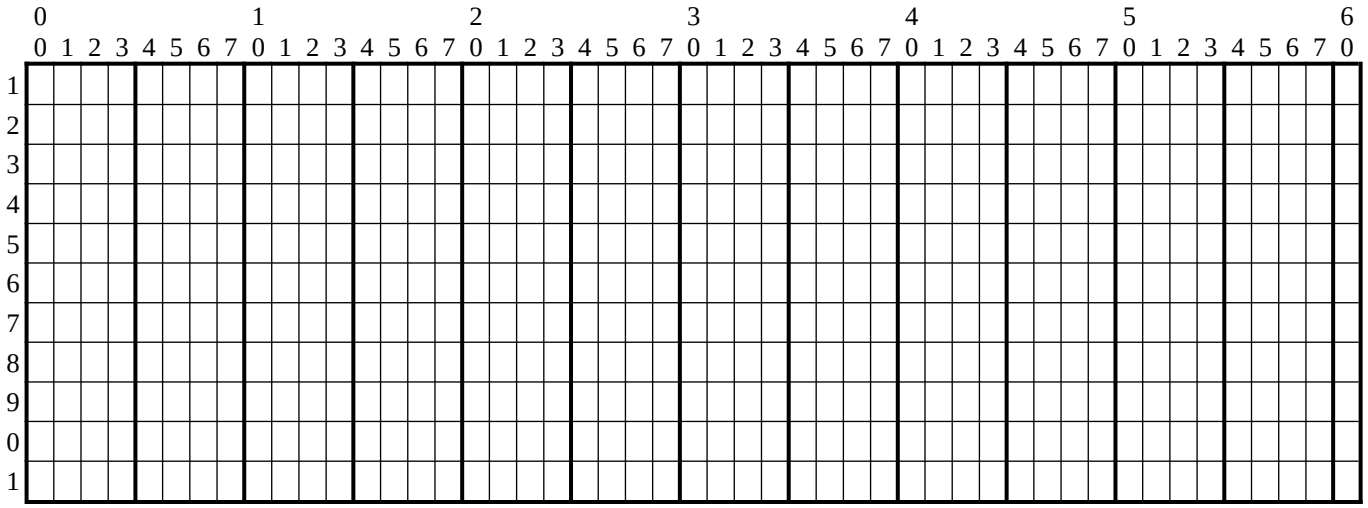


`Stella.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | 00 | 00 | 00 |
| 0000090 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000A0 | 80 | 00 | FF | 1E | 71 | C7 | 1C | 3F | 80 | 00 | FE | ED | AF | EF | BB | BF |
| 00000B0 | 80 | 00 | FF | ED | E0 | EF | BC | 3F | 80 | 00 | FF | 1D | EE | EF | BF | BF |
| 00000C0 | 80 | 00 | BA | F8 | 71 | EF | BC | 6E | 80 | 00 | D6 | ED | FF | EF | BF | F5 |
| 00000D0 | 80 | 00 | EF | 1D | FF | CF | 3F | FB | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number



# Worksheet: Reading a Bitmap File



1. Fill in the bytes of the bitmap file header according the the hexadecimal dump of your `bmp` file.

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| use | signature | | file size | | | | reserved1 | | reserved1 | | file offset to pixel array | | | |
| data | | | | | | | | | | | | | | |

| | | | | |
|---|---|---|---|---|
| 2. | a) | Write the first two bytes of the file in hexadecimal format. | | |
| | b0 | Convert these two bytes to their ASCII equivalent. | | |

| | | | |
|---|---|---|---|
| 3. | a) | Find the 4 bytes in the bitmap file header that represents the file length, and write the length of the file as an 8-digit hexadecimal number. Remember to convert from little-endian. Check that the length value corresponds correctly to the length of the file. | |

| | | | |
|---|---|---|---|
| 4. | a) | Find the 4 bytes in the bitmap file header that represents the offset of the *image data pixel array*, and write this value as an 8-digit hexadecimal number. (Again, convert from little-endian). | |

| | | | |
|---|---|---|---|
| 5. | a) | Find the start of the DIB header and write the value of the DIB header size as an 8-digit hexadecimal. | |
| | b) | Convert the value in part (a) to a decimal value and confirm that it is the correct length for a version 5 DIB header (denary 124). | |

| | | | | hexadecimal | denary |
|---|---|---|---|---|---|
| 6. | a) | Find the image height and image width fields of the DIB header and write their values – first as an 8-digit hexadecimal number, then convert to denary. | width | | |
| | | | height | | |

| | | | | |
|---|---|---|---|---|
| 7. | a) | Find the number of *bits per pixel* in the DIB header and write the value first as a 4 digit hexadecimal number, then converted to denary. | | |

| | | | |
|---|---|---|---|
| 8. | a) | Find the compression method field of the DIB header and write the value as an 8-digit hexadecimal number. | |

9. Fill in the four bytes that make up each color from the color table, then look up the HTML/CSS named color that corresponds to the RGB portion.

| **File offset** | **index** | **ARGB hexadecimal** | | | | **Named Color (HTML/CSS)** |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |

# Worksheet: Reading a Bitmap File



`Una.bmp`

| Address of Leftmost Byte | Least Significant Nibble of Address: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000000 | 42 | 4D | EA | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 92 | 00 | 00 | 00 | 7C | 00 |
| 0000010 | 00 | 00 | 31 | 00 | 00 | 00 | 0B | 00 | 00 | 00 | 01 | 00 | 01 | 00 | 00 | 00 |
| 0000020 | 00 | 00 | 58 | 00 | 00 | 00 | 13 | 0B | 00 | 00 | 13 | 0B | 00 | 00 | 02 | 00 |
| 0000030 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | E0 | 07 | 00 | 00 | 1F | 00 |
| 0000040 | 00 | 00 | 00 | 00 | 00 | 00 | 42 | 47 | 52 | 73 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 |
| 0000080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 80 | 00 | 00 | 00 | FF | FF |
| 0000090 | 00 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000A0 | 80 | 00 | C6 | EC | 3F | 82 | EC | 72 | 80 | 00 | BA | EB | BF | BE | EB | AC |
| 00000B0 | 80 | 00 | BA | EC | 3F | DE | EB | AE | 80 | 00 | BA | 6F | BF | EE | 6B | AE |
| 00000C0 | 80 | 00 | BA | 9C | 6D | F6 | 9C | 6E | 80 | 00 | BB | FF | EA | FA | FF | FF |
| 00000D0 | 80 | 00 | BB | FF | F6 | 82 | FF | FF | 80 | 00 | FF | FF | FF | FF | FF | FF |
| 00000E0 | 80 | 00 | FF | FF | FF | FF | FF | FF | 80 | 00 | | | | | | |

Byte Number