3   MIPS projects

Each project should be delivered as an *.asm file and a set of related test cases. The test cases should confirm that the program works correctly. Each project should contain a directory named *tests*. The sub directories of the directory *tests* correspond to individual tests. They should contain:

- input file(s),
- output file(s),
- a text file (named *description.txt*) containing short description what was being tested and the values of the input parameters (if any).

3.15 Binary turtle graphics – version 1

In computer graphics, turtle graphics are vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, on/off (or up/down) state [2].

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it or setting its color.

Your task is to write a program, which translates binary encoded turtle commands to a raster image in a BMP file [1].

Turtle commands

The length of all turtle commands is 16 or 32 bits. The first two bits define one of four commands (set position, set direction, move, set state). Unused bits in all commands are marked by the – character. They should not be taken into account when the command is decoded.

Set position **command**

The *set position* command sets the new coordinates of the turtle. It consists of two words. The first word defines the command (bits 15-14). The point (0,0) is located in the bottom left corner of the image. The second word contains the X (bits x9-x0) and Y (bits y5-y0) coordinates of the new position.

Table 1. The first word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
0	**0**	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Table 2. The second word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
y5	y4	y3	y2	y1	y0	x9	x8
bit no. 7	6	5	4	3	2	1	0
x7	x6	x5	x4	x3	x2	x1	x0

Set direction command

The *set direction* command sets the direction in which the turtle will move, when a move command is issued. The direction is defined by the d1, d0 bits.

Table 3. The *set direction* command.

bit no. 15	14	13	12	11	10	9	8
0	**1**	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	d1	d0

Table 4. The description of the d1,d0 bits.

Bits d1,d0	Turtle direction

00	right
01	up
10	left
11	down

Move command

The *move* command moves the turtle in direction specified by the d1-d0 bits. The movement distance is defined by the m9-m0 bits. If the destination point is located beyond the drawing area the turtle should stop at the edge of the drawing. It can't leave the drawing area. The turtle leaves a visible trail when the pen is lowered (bit ud). The color of the trail is defined by the r3-r0, g3-g0, b3-b0 bits.

Table 5. The *move* command.

bit no. 15	14	13	12	11	10	9	8
1	**0**	-	-	-	-	m9	m8
bit no. 7	6	5	4	3	2	1	0
m7	m6	m5	m4	m3	m2	m1	m0

Set pen state command

The *pen state* command defines whether the pen is raised or lowered (bit ud) and the color of the trail. Bits r3-r0 are the most significant bits of the 8-bits red component of the color (remaining bits are set to zero). Bits g3-g0 are the most significant bits of the 8-bits green component of the color (remaining bits are set to zero). Bits b3-b0 are the most significant bits of the 8-bits blue component of the color (remaining bits are set to zero).

Table 6. The *pen state* command.

bit no. 15	14	13	12	11	10	9	8
1	**1**	ud	-	b3	b2	b1	b0
bit no. 7	6	5	4	3	2	1	0
g3	g2	g1	g0	r3	r2	r1	r0

Table 7. The description of the ud bit.

ud bit	Pen state
0	pen raised (up)
1	pen lowered (down)

Input
- binary file containing 16-bits turtle commands
- file name: "input.bin"

Output
- BMP file containing the generated image:
  - Sub format: 24 bits RGB – no compression,
  - Image size: 600x50 px,
- file name: "output.bmp"

References:
[1] "**file-format-bmp**", https://en.wikipedia.org/wiki/BMP_file_format
[2] "**Turtle graphics**", https://en.wikipedia.org/wiki/Turtle_graphics
[3] Example program for bmp reading/writing,
     http://galera.ii.pw.edu.pl/~zsz/ecoar/bmp/bmp_mips.zip

3.16   Binary turtle graphics – version 2

In computer graphics, turtle graphics are vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, on/off (or up/down) state [2].

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it or setting its color.

Your task is to write a program, which translates binary encoded turtle commands to a raster image in a BMP file [1].

Turtle commands
The length of all turtle commands is 16 or 32 bits. The first two bits define one of four commands (set position, set direction, move, set state). Unused bits in all commands are marked by the – character. They should not be taken into account when the command is decoded.

Set position **command**
The *set position* command sets the new coordinates of the turtle. It consists of two words. The first word defines the command (bits 1-0) and Y (bits y5-y0) coordinate of the new position. The second word contains the X (bits x9-x0) coordinate of the new position. The point (0,0) is located in the bottom left corner of the image.

Table 1. The first word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
y5	y4	y3	y2	y1	y0	**1**	**1**

Table 2. The second word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
-	-	-	-	-	-	x9	x8
bit no. 7	6	5	4	3	2	1	0
x7	x6	x5	x4	x3	x2	x1	x0

Set direction command
The *set direction* command sets the direction in which the turtle will move, when a move command is issued. The direction is defined by the d1, d0 bits.

Table 3. The *set direction* command.

bit no. 15	14	13	12	11	10	9	8
d1	d0	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	**1**	**0**

Table 4. The description of the d1,d0 bits.

Bits d1,d0	Turtle direction

00	Right
01	Up
10	Left
11	Down

Move command

The *move* command moves the turtle in direction specified by the d1-d0 bits. The movement distance is defined by the m9-m0 bits. If the destination point is located beyond the drawing area the turtle should stop at the edge of the drawing. It can't leave the drawing area. The turtle leaves a visible trail when the pen is lowered (bit ud). The color of the trail is defined by the r3-r0, g3-g0, b3-b0 bits.

Table 5. The *move* command.

bit no. 15	14	13	12	11	10	9	8
m9	m8	m7	m6	m5	m4	m3	m2
bit no. 7	6	5	4	3	2	1	0
m1	m0	-	-	-	-	**0**	**1**

Set pen state command

The *pen state* command defines whether the pen is raised or lowered (bit ud) and the color of the trail. Bits r3-r0 are the most significant bits of the 8-bits red component of the color (remaining bits are set to zero). Bits g3-g0 are the most significant bits of the 8-bits green component of the color (remaining bits are set to zero). Bits b3-b0 are the most significant bits of the 8-bits blue component of the color (remaining bits are set to zero).

Table 6. The *pen state* command.

bit no. 15	14	13	12	11	10	9	8
r3	r2	r1	r0	g3	g2	g1	g0
bit no. 7	6	5	4	3	2	1	0
b3	b2	b1	b0	ud	-	**0**	**0**

Table 7. The description of the ud bit.

ud bit	Pen state
0	pen raised (up)
1	pen lowered (down)

Input
- binary file containing 16-bits turtle commands
- file name: "input.bin"

Output
- BMP file containing the generated image:
  - Sub format: 24 bits RGB – no compression,
  - Image size: 600x50 px,
- file name: "output.bmp"

References:
[1] "**file-format-bmp**", https://en.wikipedia.org/wiki/BMP_file_format
[2] "**Turtle graphics**", https://en.wikipedia.org/wiki/Turtle_graphics
[3] Example program for bmp reading/writing,
    http://galera.ii.pw.edu.pl/~zsz/ecoar/bmp/bmp_mips.zip

3.17 Binary turtle graphics – version 3

In computer graphics, turtle graphics are vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, on/off (or up/down) state [2].

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it or setting its color.

Your task is to write a program, which translates binary encoded turtle commands to a raster image in a BMP file [1].

Turtle commands

The length of all turtle commands is 16 or 32 bits. The first two bits define one of four commands (set position, set direction, move, set state). Unused bits in all commands are marked by the – character. They should not be taken into account when the command is decoded.

Set position **command**

The *set position* command sets the new coordinates of the turtle. It consists of two words. The first word defines the command (bits 15-14) and Y (bits y5-y0) coordinate of the new position. The second word contains the X (bits x9-x0) coordinate. The point (0,0) is located in the bottom left corner of the image.

Table 1. The first word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
1	**1**	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
y5	y4	y3	y2	y1	y0	-	-

Table 2. The second word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
x9	x8	x7	x6	x5	x4	x3	x2
bit no. 7	6	5	4	3	2	1	0
x1	x0	-	-	-	-	-	-

Set direction command

The *set direction* command sets the direction in which the turtle will move, when a move command is issued. The direction is defined by the d1, d0 bits.

Table 3. The *set direction* command.

bit no. 15	14	13	12	11	10	9	8
1	**0**	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	d1	d0

Table 4. The description of the d1,d0 bits.

Bits d1,d0	Turtle direction

00	right
01	up
10	left
11	down

Move command

The *move* command moves the turtle in direction specified by the d1-d0 bits. The movement distance is defined by the m9-m0 bits. If the destination point is located beyond the drawing area the turtle should stop at the edge of the drawing. It can't leave the drawing area. The turtle leaves a visible trail when the pen is lowered (bit ud). The color of the trail is defined by the r3-r0, g3-g0, b3-b0 bits.

Table 5. The *move* command.

bit no. 15	14	13	12	11	10	9	8
0	**1**	m9	m8	m7	m6	m5	m4
bit no. 7	6	5	4	3	2	1	0
m3	m2	m1	m0	-	-	-	-

Set pen state command

The *pen state* command defines whether the pen is raised or lowered (bit ud) and the color of the trail. Bits c2-c0 select one of the predefined colors from the color table.

Table 6. The *pen state* command.

bit no. 15	14	13	12	11	10	9	8
0	**0**	ud	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	c2	c1	c0

Table 7. The description of the ud bit.

ud bit	Pen state
0	pen raised (up)
1	pen lowered (down)

Table 8. Color table.

bits c2,c1,c0	Pen state
000	black
001	red
010	green
011	blue
100	yellow
101	cyan
110	purple
111	white

Input
- binary file containing 16-bits turtle commands
- file name: "input.bin"

Output
- BMP file containing the generated image:
    - Sub format: 24 bits RGB – no compression,
    - Image size: 600x50 px,
- file name: "output.bmp"

References:
[1] "**file-format-bmp**", https://en.wikipedia.org/wiki/BMP_file_format
[2] "**Turtle graphics**", https://en.wikipedia.org/wiki/Turtle_graphics
[3] Example program for bmp reading/writing,
    http://galera.ii.pw.edu.pl/~zsz/ecoar/bmp/bmp_mips.zip

3.18   Binary turtle graphics – version 4

In computer graphics, turtle graphics are vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, on/off (or up/down) state [2].

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it or setting its color.

Your task is to write a program, which translates binary encoded turtle commands to a raster image in a BMP file [1].

Turtle commands

The length of all turtle commands is 16 or 32 bits. The first two bits define one of four commands (set position, set direction, move, set state). Unused bits in all commands are marked by the – character. They should not be taken into account when the command is decoded.

Set position **command**

The *set position* command sets the new coordinates of the turtle. It consists of two words. The first word defines the command (bits 1-0) and X (bits x9-x0) coordinate of the new position. The second word contains the Y (bits y5-y0) coordinate of the new position. The point (0,0) is located in the bottom left corner of the image.

Table 1. The first word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
x9	x8	x7	x6	x5	x4	x3	x2
bit no. 7	6	5	4	3	2	1	0
x1	x0	-	-	-	-	**1**	**1**

Table 2. The second word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
y5	y4	y3	y2	y1	y0	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Set direction command

The *set direction* command sets the direction in which the turtle will move, when a move command is issued. The direction is defined by the d1, d0 bits.

Table 3. The *set direction* command.

bit no. 15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	d1	d0	**1**	**0**

Table 4. The description of the d1,d0 bits.

Bits d1,d0	Turtle direction

00	Right
01	Up
10	Left
11	Down

Move command

The *move* command moves the turtle in direction specified by the d1-d0 bits. The movement distance is defined by the m9-m0 bits. If the destination point is located beyond the drawing area the turtle should stop at the edge of the drawing. It can't leave the drawing area. The turtle leaves a visible trail when the pen is lowered (bit ud). The color of the trail is defined by the r3-r0, g3-g0, b3-b0 bits.

Table 5. The *move* command.

bit no. 15	14	13	12	11	10	9	8
-	-	-	-	-	-	m9	m8
bit no. 7	6	5	4	3	2	1	0
m5	m4	m3	m2	m1	m0	**0**	**1**

Set pen state command

The *pen state* command defines whether the pen is raised or lowered (bit ud) and the color of the trail. Bits c2-c0 select one of the predefined colors from the color table.

Table 6. The *pen state* command.

bit no. 15	14	13	12	11	10	9	8
c2	c1	c0	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	ud	-	**0**	**0**

Table 7. The description of the ud bit.

ud bit	Pen state
0	pen raised (up)
1	pen lowered (down)

Table 8. Color table.

bits c2,c1,c0	Pen state
000	black
001	red
010	green
011	blue
100	yellow
101	cyan
110	purple
111	white

Input
- binary file containing 16-bits turtle commands
- file name: "input.bin"

Output
- BMP file containing the generated image:
  - Sub format: 24 bits RGB – no compression,
  - Image size: 600x50 px,
- file name: "output.bmp"

References:
[1] "**file-format-bmp**", https://en.wikipedia.org/wiki/BMP_file_format
[2] "**Turtle graphics**", https://en.wikipedia.org/wiki/Turtle_graphics
[3] Example program for bmp reading/writing,
      http://galera.ii.pw.edu.pl/~zsz/ecoar/bmp/bmp_mips.zip

3.19   Binary turtle graphics – version 5

In computer graphics, turtle graphics are vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, on/off (or up/down) state [2].

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it or setting its color.

Your task is to write a program, which translates binary encoded turtle commands to a raster image in a BMP file [1].

Turtle commands

The length of all turtle commands is 16 or 32 bits. The first two bits define one of four commands (set position, set direction, move, set state). Unused bits in all commands are marked by the – character. They should not be taken into account when the command is decoded.

Set position command

The *set position* command sets the new coordinates of the turtle. It consists of two words. The first word defines the command (bits 15-14). The point (0,0) is located in the bottom left corner of the image. The second word contains the X (bits x9-x0) and Y (bits y5-y0) coordinates of the new position.

Table 1. The first word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
0	**1**	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Table 2. The second word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
y5	y4	y3	y2	y1	y0	x9	x8
bit no. 7	6	5	4	3	2	1	0
x7	x6	x5	x4	x3	x2	x1	x0

Set direction command

The *set direction* command sets the direction in which the turtle will move, when a move command is issued. The direction is defined by the d1, d0 bits.

Table 3. The *set direction* command.

bit no. 15	14	13	12	11	10	9	8
0	**0**	-	-	-	-	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	d1	d0

Table 4. The description of the d1,d0 bits.

Bits d1,d0	Turtle direction

00	up
01	left
10	down
11	right

Move command

The *move* command moves the turtle in direction specified by the d1-d0 bits. The movement distance is defined by the m9-m0 bits. If the destination point is located beyond the drawing area the turtle should stop at the edge of the drawing. It can't leave the drawing area. The turtle leaves a visible trail when the pen is lowered (bit ud). The color of the trail is defined by the r3-r0, g3-g0, b3-b0 bits.

Table 5. The *move* command.

bit no. 15	14	13	12	11	10	9	8
1	**1**	-	-	-	-	m9	m8
bit no. 7	6	5	4	3	2	1	0
m7	m6	m5	m4	m3	m2	m1	m0

Set pen state command

The *pen state* command defines whether the pen is raised or lowered (bit ud) and the color of the trail. Bits r3-r0 are the most significant bits of the 8-bits red component of the color (remaining bits are set to zero). Bits g3-g0 are the most significant bits of the 8-bits green component of the color (remaining bits are set to zero). Bits b3-b0 are the most significant bits of the 8-bits blue component of the color (remaining bits are set to zero).

Table 6. The *pen state* command.

bit no. 15	14	13	12	11	10	9	8
1	**0**	-	ud	b3	b2	b1	b0
bit no. 7	6	5	4	3	2	1	0
g3	g2	g1	g0	r3	r2	r1	r0

Table 7. The description of the ud bit.

ud bit	Pen state
0	pen lowered (down)
1	pen raised (up)

Input
- binary file containing 16-bits turtle commands
- file name: "input.bin"

Output
- BMP file containing the generated image:
  - Sub format: 24 bits RGB – no compression,
  - Image size: 600x50 px,
- file name: "output.bmp"

References:
[1] "**file-format-bmp**", https://en.wikipedia.org/wiki/BMP_file_format
[2] "**Turtle graphics**", https://en.wikipedia.org/wiki/Turtle_graphics
[3] Example program for bmp reading/writing,
        http://galera.ii.pw.edu.pl/~zsz/ecoar/bmp/bmp_mips.zip

3.20   Binary turtle graphics – version 6

In computer graphics, turtle graphics are vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, on/off (or up/down) state [2].

The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it or setting its color.

Your task is to write a program, which translates binary encoded turtle commands to a raster image in a BMP file [1].

Turtle commands

The length of all turtle commands is 16 or 32 bits. The first two bits define one of four commands (set position, set direction, move, set state). Unused bits in all commands are marked by the – character. They should not be taken into account when the command is decoded.

Set position command

The *set position* command sets the new coordinates of the turtle. It consists of two words. The first word defines the command (bits 15-14) and Y (bits y5-y0) coordinate of the new position. The second word contains the X (bits x9-x0) coordinate. The point (0,0) is located in the bottom left corner of the image.

Table 1. The first word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
1	**0**	y5	y4	y3	y2	y1	y0
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Table 2. The second word of the *set position* command.

bit no. 15	14	13	12	11	10	9	8
x9	x8	x7	x6	x5	x4	x3	x2
bit no. 7	6	5	4	3	2	1	0
x1	x0	-	-	-	-	-	-

Set direction command

The *set direction* command sets the direction in which the turtle will move, when a move command is issued. The direction is defined by the d1, d0 bits.

Table 3. The *set direction* command.

bit no. 15	14	13	12	11	10	9	8
1	**1**	-	-	d1	d0	-	-
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Table 4. The description of the d1,d0 bits.

Bits d1,d0	Turtle direction
00	up
01	left
10	down
11	right

Move command

The *move* command moves the turtle in direction specified by the d1-d0 bits. The movement distance is defined by the m9-m0 bits. If the destination point is located beyond the drawing area the turtle should stop at the edge of the drawing. It can't leave the drawing area. The turtle leaves a visible trail when the pen is lowered (bit ud). The color of the trail is defined by the r3-r0, g3-g0, b3-b0 bits.

Table 5. The *move* command.

bit no. 15	14	13	12	11	10	9	8
0	**0**	m9	m8	m7	m6	m5	m4
bit no. 7	6	5	4	3	2	1	0
m3	m2	m1	m0	-	-	-	-

Set pen state command

The *pen state* command defines whether the pen is raised or lowered (bit ud) and the color of the trail. Bits c2-c0 select one of the predefined colors from the color table.

Table 6. The *pen state* command.

bit no. 15	14	13	12	11	10	9	8
0	**1**	-	ud	-	c2	c1	c0
bit no. 7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Table 7. The description of the ud bit.

ud bit	Pen state
0	pen lowered (down)
1	pen raised (up)

Table 8. Color table.

bits c2,c1,c0	Pen state
000	black
001	purple
010	cyan
011	yellow
100	blue
101	green
110	red
111	white

Input
- binary file containing 16-bits turtle commands
- file name: "input.bin"

Output
- BMP file containing the generated image:
  - Sub format: 24 bits RGB – no compression,
  - Image size: 600x50 px,
- file name: "output.bmp"

References:
[1] "**file-format-bmp**", https://en.wikipedia.org/wiki/BMP_file_format
[2] "**Turtle graphics**", https://en.wikipedia.org/wiki/Turtle_graphics
[3] Example program for bmp reading/writing,
    http://galera.ii.pw.edu.pl/~zsz/ecoar/bmp/bmp_mips.zip

The Structure of the Bitmap Image File (BMP)

Bitmap File Header
BITMAPFILEHEADER
- Signature
- File Size
- Reserved1 | Reserved2
- File Offset to PixelArray

DIB Header
BITMAPV5HEADER
- DIB Header Size
- Image Width (w)
- Image Height (h)
- Planes | Bits per Pixel
- Compression
- Image Size
- X Pixels Per Meter
- Y Pixels Per Meter
- Colors in Color Table
- Important Color Count
- Red channel bitmask
- Green channel bitmask
- Blue channel bitmask
- Alpha channel bitmask
- Color Space Type
- Color Space Endpoints
- Gamma for Red channel
- Gamma for Green channel
- Gamma for Blue channel
- Intent
- ICC Profile Data
- ICC Profile Size
- Reserved

Older DIB Headers can be substituted for the **BITMAPV5HEADER**

Note: The size of Color Space Endpoints is 36 Bytes

(This diagram does not depict it proportionally. It is drawn in that manner only to save vertical space.)

Color Table (semi-optional)
- Color Definition (index 0)
- Color Definition (index 1)
- Color Definition (index 2)
- Color Definition (index n)

Note: The presence of the Color Table is mandatory when Bits per Pixel ≤ 8

Note: The size of Color Table Entries is 3 Bytes if **BITMAPCOREHEADER** is substituted for **BITMAPV5HEADER**.

GAP1 (optional)

Pixel Format

Pad row size to a multiple of 4 Bytes

Image Data
PixelArray [x,y]

Pixel[0,h-1]	Pixel[1,h-1]	Pixel[2,h-1]	...	Pixel[w-1,h-1]	Padding
Pixel[0,h-2]	Pixel[1,h-2]	Pixel[2,h-2]	...	Pixel[w-1,h-2]	Padding
Pixel[0,9]	Pixel[1,9]	Pixel[2,9]	...	Pixel[w-1,9]	Padding
Pixel[0,8]	Pixel[1,8]	Pixel[2,8]	...	Pixel[w-1,8]	Padding
Pixel[0,7]	Pixel[1,7]	Pixel[2,7]	...	Pixel[w-1,7]	Padding
Pixel[0,6]	Pixel[1,6]	Pixel[2,6]	...	Pixel[w-1,6]	Padding
Pixel[0,5]	Pixel[1,5]	Pixel[2,5]	...	Pixel[w-1,5]	Padding
Pixel[0,4]	Pixel[1,4]	Pixel[2,4]	...	Pixel[w-1,4]	Padding
Pixel[0,3]	Pixel[1,3]	Pixel[2,3]	...	Pixel[w-1,3]	Padding
Pixel[0,2]	Pixel[1,2]	Pixel[2,2]	...	Pixel[w-1,2]	Padding
Pixel[0,1]	Pixel[1,1]	Pixel[2,1]	...	Pixel[w-1,1]	Padding
Pixel[0,0]	Pixel[1,0]	Pixel[2,0]	...	Pixel[w-1,0]	Padding

GAP2 (optional)

ICC Color Profile (optional)
Embedded, variable size ICC Color Profile Data.

(or path to a **linked** file containing ICC Color Profile Data)

Note: The ICC Color Profile may be present only when the **BITMAPV5HEADER** is used.

(This diagram wrongly suggests that the size of Color Profile must be a multiple of 4 Bytes. It is drawn in that manner only to save vertical space.)

Byte offset: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |