PPM IMAGES

CPSC 1010 Fall 2015

OVERVIEW

- Stands for Portable Pixel Map
- They have a simple format
- You will need a specific viewer to view ppm
- Gimp works for Windows, Mac, or Linux (little slow)
  - I will use gimp to test your programs
- IrfanView (www.irfanview.com) works for windows as well
- Images contain two important things
  - header
  - group of pixels
    - each pixel has three values RGB (red, green, and blue)

LITTLE MORE ABOUT IMAGES

- There are various type of image format; we will concentrate on PPM
  - PPM supports full-color images. Reads either pbm, pgm, or ppm formats, writes ppm format
- Couple others in the same family as PPM:
  - PGM - supports greyscale images
    - reads either pbm or pgm format and writes pgm format
  - PBM - supports monochrome bitmaps (one bit per pixel)

PPM HEADERS

EXAMPLES OF HEADERS

P6 1024 788 255

or

P6

400 400 255

or

P6

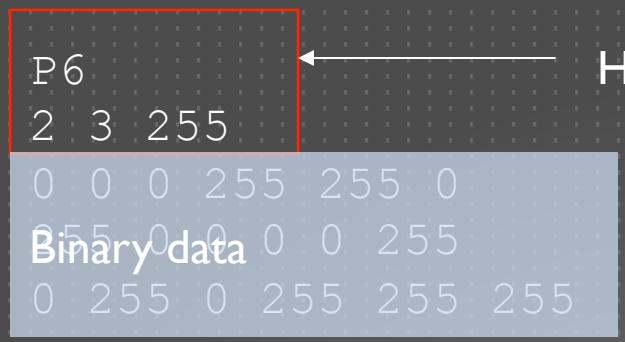
#This is a comment

400 400 #This is another comment

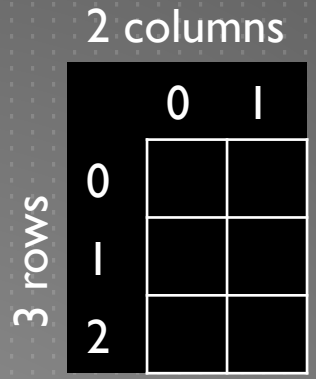
255

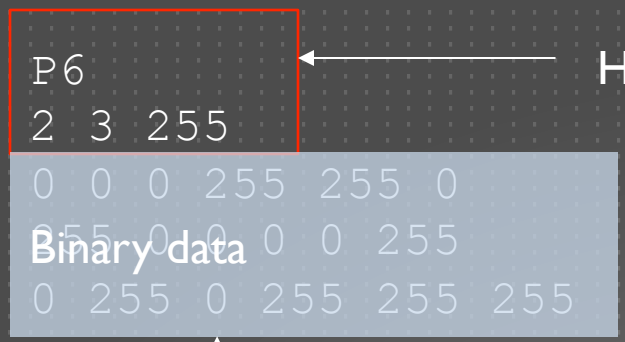
You can add comments using #comment

DO NOT PUT A SPACE AFTER 255



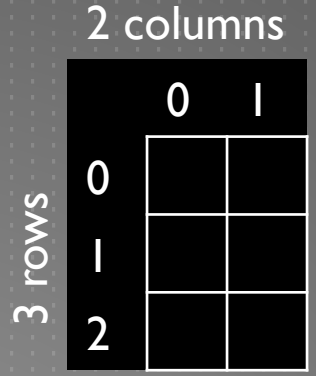
P6: tells the computer it is a ppm file
2: tells the computer the width (columns) of the image is 2 pixels
3: tells the computer the height (rows) of the image is 3 pixels
255: tells the computer the maximum color value (this is always 255)

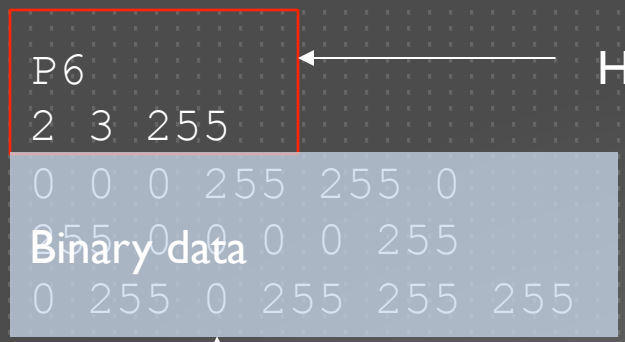




The remainder of the file is binary data

P6: tells the computer it is a ppm file
2: tells the computer the width (columns) of the image is 2 pixels
3: tells the computer the height (rows) of the image is 3 pixels
255: tells the computer the maximum color value (this is always 255)



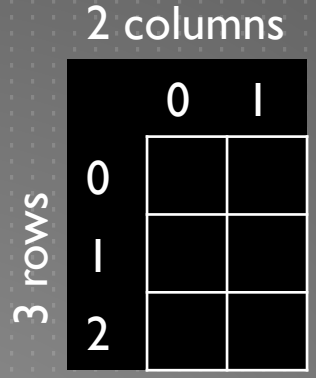


P6: tells the computer it is a ppm file
2: tells the computer the width (columns) of the image is 2 pixels
3: tells the computer the height (rows) of the image is 3 pixels
255: tells the computer the maximum color value (this is always 255)

Conceptually, you can think of the binary data as a consecutive set of bytes

Each set of three bytes represents a pixel

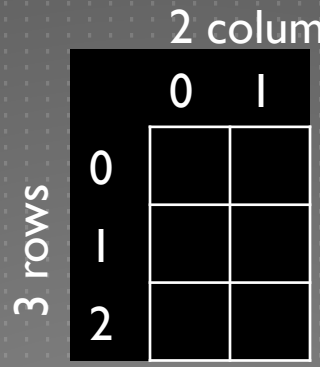
Each pixel represents a color channel (RGB)



```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```

If I were to "decode", each set of three would represent a pixel, with the bytes in this order:

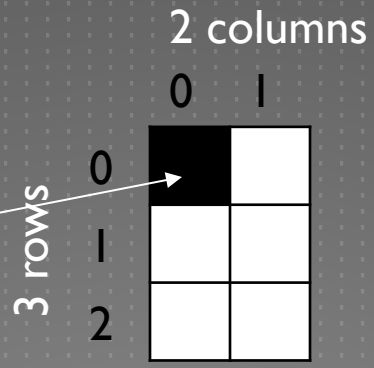
Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)



Each set of three after that determines the color of a pixel in this order:

Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)

```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```

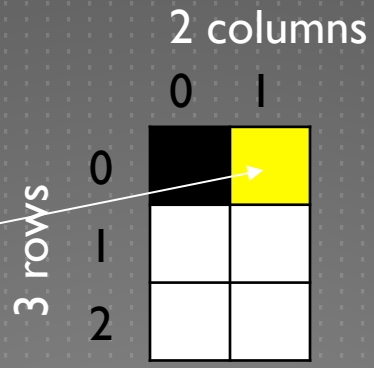


0 red, 0 green, 0 blue = black

Each set of three after that determines the color of a pixel in this order:

Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)

```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```

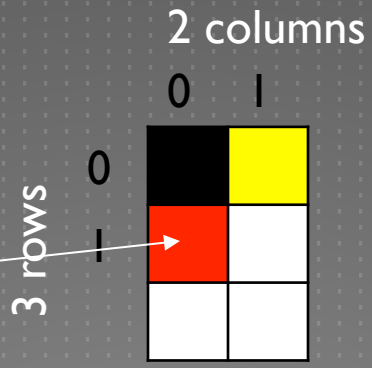


255 red, 255 green, 0 blue = yellow

Each set of three after that determines the color of a pixel in this order:

Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)

```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```

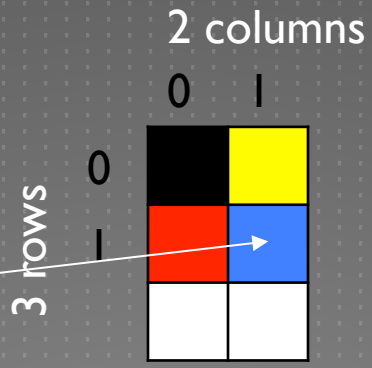


255 red, 0 green, 0 blue = red

Each set of three after that determines the color of a pixel in this order:

Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)

```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```



0 red, 0 green, 255 blue = blue

Each set of three after that determines the color of a pixel in this order:

Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)

```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```

0 red, 255 green, 0 blue = green

Each set of three after that determines the color of a pixel in this order:

Amount red (0-255)
Amount green (0-255)
Amount blue (0-255)

```
P6
2 3 255
0 0 0 255 255 0
255 0 0 0 0 255
0 255 0 255 255 255
```

255 red, 255 green, 255 blue = white

LET'S LOOK AT SOME CODE

- (ppm_fun.c)

- How many bytes does it take to create a 400 X 400 image?

- convert ppm to jpg
- in terminal type:

convert filename.ppm filename.jpg

This should work in linux and mac not sure about windows

COUPLE COOL PROCEDURALLY GENERATED PPM IMAGES

- http://people.cs.clemson.edu/~chochri/101/Modules/ppm/julia.jpg
- http://www.physics.emory.edu/faculty/weeks//pics/software/lcca.c
- http://people.cs.clemson.edu/~chochri/101/Modules/ppm/mandelbrot.jpg