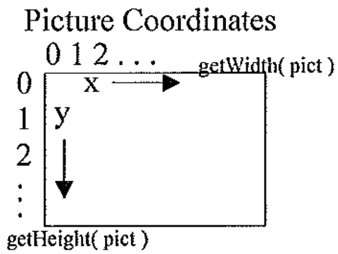
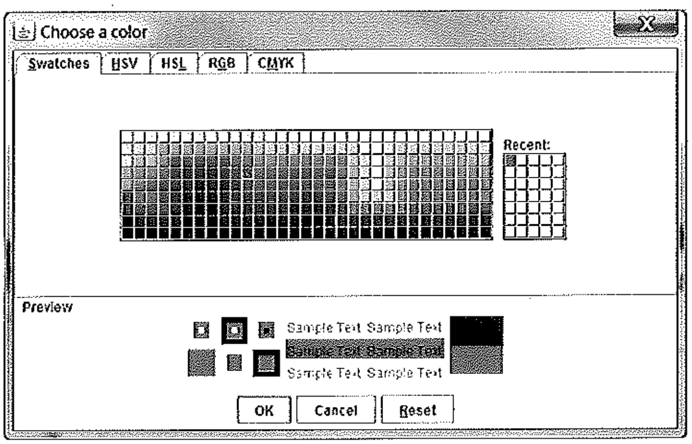
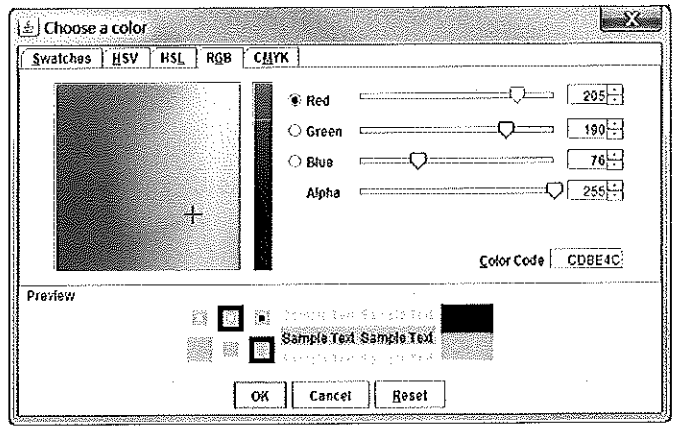
 Name:________________

Pictures are encoded as a 2D array/matrix of "dots"/*pixels* (i.e., picture elements). With enough pixels human beings cannot distinguish individual "dots." Visible light is a continuous spectrum of wavelengths between 370 nm to 730 nm. However, human brains determine color based on feedback from three sensors that peak around 425 nm (blue), 550 nm (green), and 560 nm (red), so we perceive for instance two kind of orange: spectral orange (wavelength for orange) and the mixture of R, B, G sensors for orange. There are several color encoding schemes, but JES mainly uses RGB with 8-bits (0 - 255_{10}) for each *channel* for 24-bits per pixel. JES's `pickAColor()` command displays:



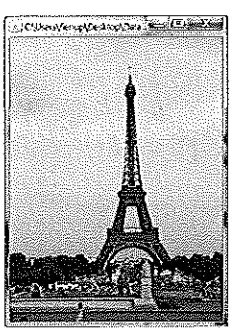
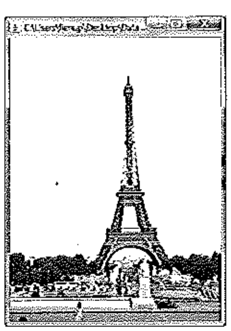
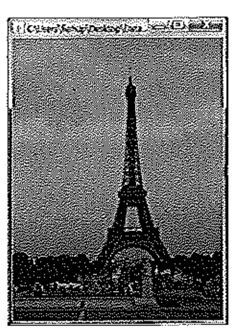
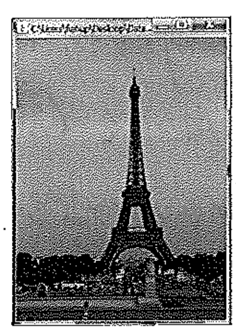
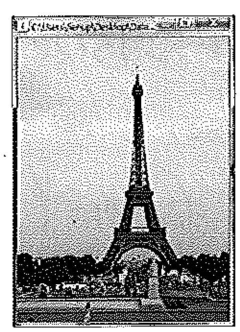
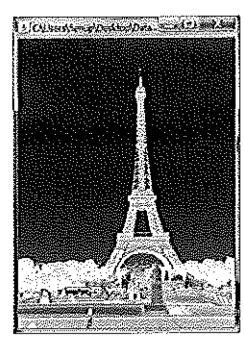




1. What are the RGB values for each of the following?

Color	R value	G value	B value
white	255	255	255
black	0	0	0
red	255	0	0
green	0	255	0
blue	0	0	255
gray	128	128	128
"light" gray lightGray	192	192	192
"dark" gray darkGray	64	64	64
~~"panther"~~ purple	Revisit In Lab on Wed.		
~~"panther" gold~~			

2. In chapter 3, we are interested in *filters*, i.e., manulating all of the pixels of a picture similarly (e.g., lighten/darken, color to grayscale, negate).

eiffel.jpg lightenFilter.py darkenFilter.py sunsetFilter.py simpleGrayScaleFilter.py negateFilter.py

a) What RGB modifications do you think were make at each pixel by the filter programs?

increase RGB values | decrease RGB values | increase R | average R,G,B + set each RGB it average | newR=255-oldR newG = 255-oldG ''

```
""" 'makeSunset' filter function described in chapter 3 """
def main():
  print "Select the Media Folder"
  setMediaFolder()
  print "Select the picture (.jpg) file"
  fileName = pickAFile()
  pict = makePicture(fileName)
  show(pict)

  print "Please wait while picture is processed."
  makeSunset(pict)

  repaint(pict)  # updates the picture shown

def makeSunset(pict):
  for px in getPixels(pict):
    reducedBlue = getBlue(px)*0.7
    reducedGreen = getGreen(px)*0.7
    originalRed = getRed(px)
    newColor = makeColor(originalRed, reducedGreen, reducedBlue)
    setColor(px, newColor)

main() # starts the program
```

3. To modify every pixel in the picture we used a `for`-loop with the general *syntax* (i.e., structure):

```
for <variable> in <sequence>:
   statement1
   statement2
   statement3
```

A `for`-loop's *semantics* (i.e., meaning) performs the indented statements (the loop *body*) once for each item in the sequence with the *loop-control variable* being assigned the item's value.

In the program `getPixels(pict)` returns a 1D array (a JES sequence) containing all the pixels of the picture passing in the `pict` parameter.

a) What is the loop-control variable's name? px

b) What type of object is the loop-control variable? pixel

c) Conjecture why reducing the blue and green RGB components of every pixel makes the image "redder."

Relative R, G, B value has more red compared to decreased red and blue components.

d) Complete the `darkenByAmt` function below that takes as a parameter an `amtToDarkenBy` used to darken each RGB component.

expect fraction between 0 and 1

```
def darkenByAmt(pict, amtToDarkenBy):
  for px in getPixels:
    darkerRed = getRed(px)*amtToDarkenBy
    darkerGreen = getGreen(px)*amtToDarkenBy
    darkerBlue = getBlue(px)*amtToDarkenBy
    darkerColor = makeColor(darkerRed, darkerGreen, darkerBlue)
    setColor(px, darkerColor)
```