```
def negative(picture):
    for px in getPixels(picture):
        red=getRed(px)
        green=getGreen(px)
        blue=getBlue(px)
        negColor=makeColor( 255-red, 255-green, 255-blue)
        setColor(px,negColor)

def decreaseRed(picture):
    for p in getPixels(picture):
    value=getRed(p)
    setRed(p,value*0.5)
```

## 3.5.3 Blending Pictures

When we create collages by copying, any overlap typically means that one picture shows *over* another. The last picture painted on is the one that appears on top of the other. But it doesn't have to be that way. We can *blend* pictures by multiplying their colors and adding them. This gives us the effect of *transparency*.

We know that 100% of something is the whole thing. 50% of one and 50% of another also is a whole. In the recipe below, we blend a picture of the mother and the daughter with an overlap of some 70 (the width of Barbara minus 150) columns of pixels (Figure 3.33).



Recipe 30: Blending two pictures

```
def blendPictures():
 barb = makePicture(getMediaPath("barbara.jpg"))
 katie = makePicture(getMediaPath("Katie-smaller.jpg"))
  canvas = makePicture(getMediaPath("640x480.jpg"))
  #Copy first 150 columns of Barb
  sourceX=1
  for targetX in range(1,150):
   sourceY=1
   for targetY in range(1,getHeight(barb)):
      color = getColor(getPixel(barb,sourceX,sourceY))
      setColor(getPixel(canvas,targetX,targetY),color)
      sourceY = sourceY + 1
    sourceX = sourceX + 1
  #Now, grab the rest of Barb at 50 overlap = getWidth(barb)-150
  sourceX=1
  for targetX in range(150,getWidth(barb)):
    sourceY=1
    for targetY in range(1,getHeight(katie)):
```

 $\overline{\mathbf{A}}$ 

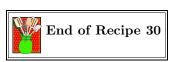


## 92 Chapter 3 Encoding and Manipulating Pictures



FIGURE 3.33: Blending the picture of mom and daughter

```
bpixel = getPixel(barb,sourceX+150,sourceY)
    kpixel = getPixel(katie,sourceX,sourceY)
    newred= 0.50*getRed(bpixel)+0.50*getRed(kpixel)
   newgreen=0.50*getGreen(bpixel)+0.50*getGreen(kpixel)
   newblue = 0.50*getBlue(bpixel)+0.50*getBlue(kpixel)
    color = makeColor(newred,newgreen,newblue)
    setColor(getPixel(canvas,targetX,targetY),color)
   sourceY = sourceY + 1
  sourceX = sourceX + 1
# Last columns of Katie
sourceX=overlap
for targetX in range(150+overlap,150+getWidth(katie)):
  sourceY=1
  for targetY in range(1,getHeight(katie)):
    color = getColor(getPixel(katie,sourceX,sourceY))
    setColor(getPixel(canvas,targetX,targetY),color)
   sourceY = sourceY + 1
  sourceX = sourceX + 1
show(canvas)
return canvas
```



## 3.5.4 Rotation

Transformations to the image occur by using the index variables differently or incrementing them differently, but otherwise keeping the same recipe. Let's rotate

 $\overline{\mathbf{A}}$