

```

public Picture blend(Picture pic1, Picture pic2)
{
    Picture merger = new Picture(pic1.getWidth() + pic1.getWidth()/2,
pic1.getHeight());
    Pixel pixel1 = null;
    Pixel pixel2 = null;
    Pixel targetPixel = null;

    int sourceX = 0;
    int targetX = 0;

    for(; sourceX < pic1.getWidth()/2; sourceX++, targetX++)
    {
        for(int sourceY = 0, targetY = 0; sourceY < pic1.getHeight();
sourceY++, targetY++)
        {
            pixel1 = pic1.getPixel(sourceX, sourceY);
            targetPixel = merger.getPixel(targetX, targetY);
            targetPixel.setColor(pixel1.getColor());
        }
    }

    for(; sourceX < pic1.getWidth(); sourceX++, targetX++)
    {
        for(int sourceY = 0, targetY = 0; sourceY < pic1.getHeight();
sourceY++, targetY++)
        {
            pixel1 = pic1.getPixel(sourceX, sourceY);
            pixel2 = pic2.getPixel(sourceX - pic1.getWidth()/2, sourceY);
            targetPixel = merger.getPixel(targetX, targetY);
            targetPixel.setColor( new Color( (int) (pixel1.getRed()*0.5 +
pixel2.getRed() *0.5),
                                                    (int) (pixel1.getGreen()*0.5 +
pixel2.getGreen()*0.5),
                                                    (int) (pixel1.getBlue()*0.5 +
pixel2.getBlue()*0.5)));
        }
    }

    sourceX = pic2.getWidth()/2;
    for(; sourceX < pic2.getWidth(); sourceX++, targetX++)
    {
        for(int sourceY = 0, targetY = 0; sourceY < pic2.getHeight();
sourceY++, targetY++)
        {
            pixel2 = pic2.getPixel(sourceX, sourceY);
            targetPixel = merger.getPixel(targetX, targetY);
            targetPixel.setColor(pixel2.getColor());
        }
    }

    return merger;
}

```

```
public Picture copyLeftRotation(Picture pic)
{
    Picture rotated = new Picture(pic.getHeight(), pic.getWidth());
    Pixel sourcePixel = null;
    Pixel targetPixel = null;

    for(int sourceX = 0; sourceX < pic.getWidht(); sourceX++)
    {
        for(int sourceY = 0; sourceY < pic.getHeight(); sourceY++)
        {
            sourcePixel = pic.getPixel(sourceX, sourceY);
            targetPixel = rotated.getPixel(sourceY, pic.getWidth() - 1 -
sourceX);
            targetPixel.setColot(sourcePixel.getColor());
        }
    }

    return rotated;
}
```

```
public Picture shrink(Picture pic)
{
    Picture smaller = new Picture(pic.getWidth()/2, pic.getHeight()/2);
    Pixel sourcePixel = null;
    Pixel targetPixel = null;

    for(int sourceX = 0, targetX = 0; sourcX < pic.getWidth(); sourceX+=2,
targetX++)
    {
        for(int sourceY = 0, targetY = 0; sourceY < pic.getHeight();
sourceY+=2, targetY++)
        {
            sourcePixel = pic.getPixel(sourceX, sourceY);
            targetPixel = smaller.getPixel(targetX, targetY);
            targetPixel.setColor(sourcePixel.getColor());
        }
    }

    return smaller;
}
```

```
public Picture enlarge(Picture pic)
{
    Picture bigger = new Picture(pic.getWidth()*2, pic.getHeight()*2);
```

```
Pixel sourcePixel = null;
Pixel targetPixel = null;

for(double sourceX = 0, targetX = 0; sourceX < pic.getWidht();
sourceX+0.5, targetX++)
{
    for(double sourceY = 0, targetY = 0; sourceY < pic.getHeight();
sourceY+=0.5, targetY++)
    {
        sourcePixel = pic.getPixel((int)sourceX, (int)sourceY);
        targetPixel = bigger.getPixel((int)targetX, (int)targetY);
        targetPixel.setColor(sourcePixel.getColor());
    }
}

return bigger;
}
```