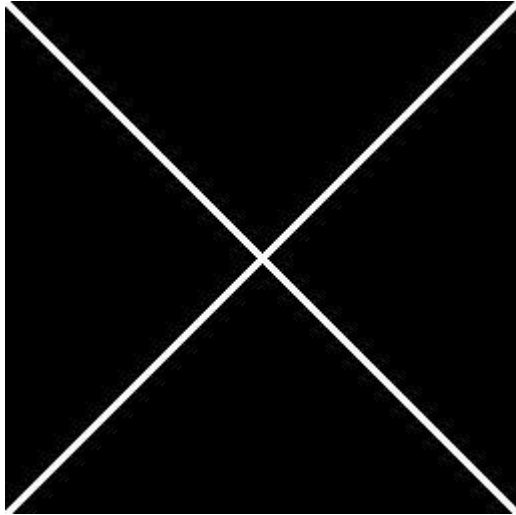


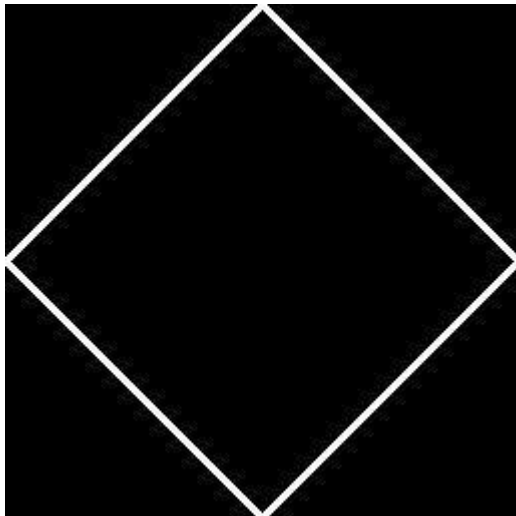
Задача 1. Графический миксер

Напишите функцию `twist_image(input_file_name, output_file_name)`, которая будет менять местами левую и правую половины изображения. Параметр `input_file_name` задаёт имя исходного файла, а `output_file_name` — имя файла, куда следует сохранить результат.

При передаче данного изображения:



должно получиться:



Примечание. Все имена файлов имеют расширение `.jpg`.

Задача 2. Прозрачность

Два изображения можно совместить так, что одно из них будет как бы «полупрозрачным». Для этого значения цветовых компонент каждого совмещаемого пикселя обоих изображений нужно попарно суммировать с определёнными весовыми коэффициентами.

Например, если итоговый цвет (в нотации RGB) вычислять по формуле:

$$R = 0.8 * R1 + 0.2 * R2$$

$$G = 0.8 * G1 + 0.2 * G2$$

$$B = 0.8 * B1 + 0.2 * B2$$

то получится, что первое изображение будет иметь 20 процентов прозрачности.

Продемонстрируем эффект прозрачности:







Напишите функцию **transparency(filename1, filename2)**, которая, используя PIL, смешивает изображения в пропорции 50/50.

Формат ввода

```
transparency("image1.jpg", "image2.jpg")
```

Формат вывода

Результат смешивания изображений сохраните в файл с именем **res.jpg** в той же папке, где находится ваша программа.

Примечания

Результат вычисления каждой компоненты нового пикселя необходимо приводить к **целому** числу с помощью функции **int**.

Например, так:

```
r = int(0.2 * r1 + 0.8 * r2)
```


Задача 3. Поворот с размытием

В файле **image.jpg** лежит изображение. Напишите функцию **motion_blur(n)** которая:

1. открывает изображение **image.jpg**
2. поворачивает его на 270 градусов против часовой стрелки без использования циклов (только встроенными средствами PIL)
3. обрабатывает полученное изображение с помощью размытия Гаусса (**GaussianBlur**) с параметром **n**
4. сохраняет результат в файле **res.jpg**

Формат ввода

Изображение **image.jpg** в текущей папке



вызов функции:
motion_blur(10)

Формат вывода

Изображение **res.jpg** в текущей папке



Задача 4. Миниатюра для сайта

В файле **image.jpg** лежит изображение. Напишите функцию **make_preview(size, n_colors)** которая:

1. открывает изображение **image.jpg**
2. уменьшает его до размера **size**
3. уменьшает число цветов в изображении до **n_colors** (такая процедура называется квантование или **quantize**)
4. сохраняет результат в файле **res.bmp**

Формат ввода

Изображение **image.jpg** в текущей папке.



вызов функции:

```
make_preview((400, 200), 64)
```

Формат вывода



Примечания

Обратите внимание, что результат сохраняется в формате **vmp**.