Laboratory 3

Pikovets Artem KM-22

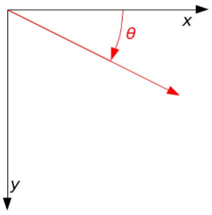
1. **We are going to use:**
   - numpy library for multiplying by transformation matrix
   - PIL library for creating the image

```
import numpy as np
from PIL import Image, ImageDraw
```

2. Get the coordinates from dataset

```
xs = []; ys = [];
with open('../data/DS8.txt', 'r') as f:
  for line in f:
    y, x = tuple(map(int, line.split()))
    xs.append(x)
    ys.append(y)
xs = np.array(xs)
ys = np.array(ys)
```

3. Create the rotation matrix in non-standard axes



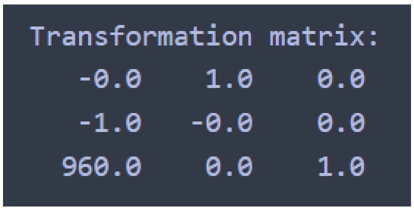
non-standard axes

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_1 & -y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_1 & y_1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ x_1 \cdot (1-\cos\theta) + y_1 \cdot \sin\theta & y_1 \cdot (1-\cos\theta) - x_1 \cdot \sin\theta & 1 \end{bmatrix}$$

x1 = y1 = 480; teta = 90º

```
alpha = 1.571  # Rotation angle
# Get the rotation matrix
rotation_matrix = np.array([
    [np.cos(alpha), np.sin(alpha), 0],
    [-np.sin(alpha), np.cos(alpha), 0],
    [480 * (1 - np.cos(alpha)) + 480 * np.sin(alpha), 480 * (1 - np.cos(alpha)) - 480 * np.sin(alpha), 1]
])
```



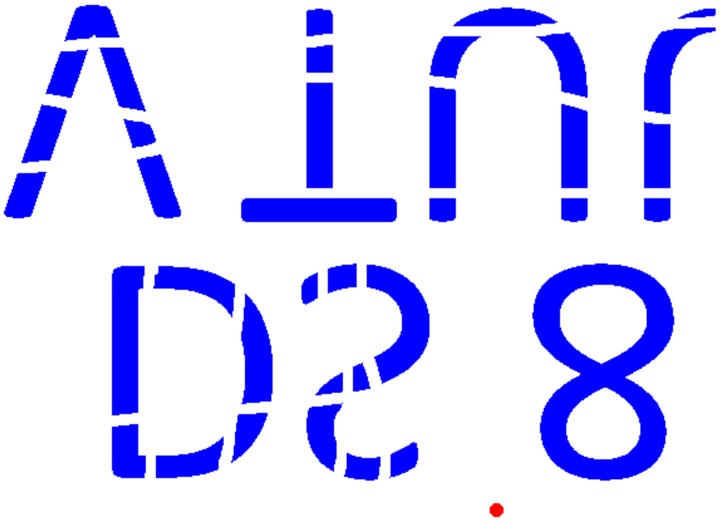
```
Transformation matrix:
    -0.0     1.0     0.0
    -1.0    -0.0     0.0
   960.0     0.0     1.0
```

4. Transform the coordinates via multiplying them by transformation matrix

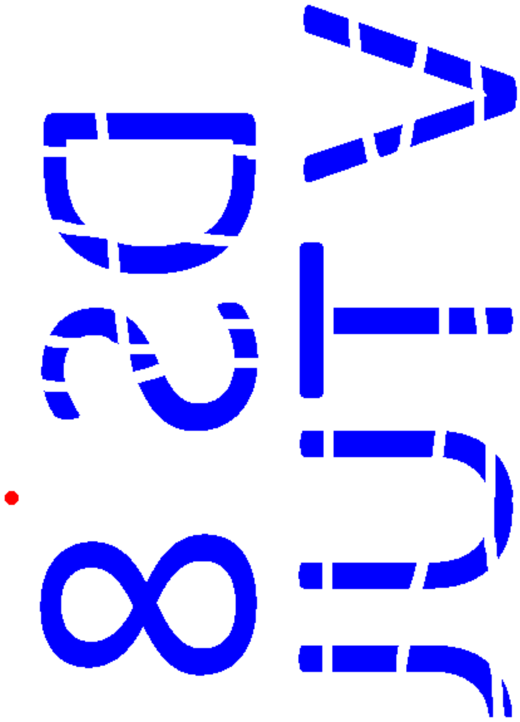
```
# Add the column of 1s; Put the points in (point_cnt, 3) shape
points = np.array([xs, ys, np.ones_like(xs)]).transpose()

# Rotate the points
points_rotated = points @ rotation_matrix
```

5. **Draw the points on the image via ImageDraw library**

Show ***default*** image (with rotation point - red circle):



Show ***rotated*** image (with rotation point - red circle):



6. Save images

```
img.save("default_image.png")
img_rotated.save("rotated_image.png")
```