```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon, Rectangle, Wedge
import cv2
from google.colab.patches import cv2_imshow

# Function to create and save the shape
def create_shape(r):
    fig, ax = plt.subplots()

    # Create the square at the center
    square = Rectangle((-r, -r), 2*r, 2*r, color='black') #from (0,0) as
origin drwaing square with 2r side
    ax.add_patch(square)

    # Create the semi-circles at the top and bottom
    top_wedge = Wedge((0, r), r, 0, 180, color='black') # 0 to 180 degrees
    bottom_wedge = Wedge((0, -r), r, 180, 360, color='black') # 180 to 360
degrees
    ax.add_patch(top_wedge)
    ax.add_patch(bottom_wedge)

    # Create the right-angled triangles at the sides
    left_triangle = Polygon([[-r, r], [-2*r, 0], [-r, -r]], color='black')
# -2r is left extreme point of the traingle
    right_triangle = Polygon([[r, r], [r, -r], [2*r, 0]], color='black') #
2r is right extreme point of the traingle
    ax.add_patch(left_triangle)
    ax.add_patch(right_triangle)

    # Create the empty square inside
    inner_square = Rectangle((-0.5*r, -0.5*r), r, r, color='white')  #from
(0,0) as origin empty square with r side
    ax.add_patch(inner_square)

    # Set the limits and aspect ratio
    ax.set_xlim(-2*r -5, 2*r + 5) # Increasing the limits to fit the shape
    ax.set_ylim(-2*r -5, 2*r + 5)  # Increasing the limits to fit the
shape
    ax.set_aspect('equal')
```

```
    # Remove axis
    ax.axis('off')

    # Save the figure
    plt.savefig('shape.png', bbox_inches='tight', pad_inches=0)
    plt.close(fig)

# Create the shape and save it
create_shape(r=10)

# Read the image file
img = cv2.imread('shape.png', cv2.IMREAD_GRAYSCALE)  # Ensure it's read in
grayscale

# Convert to binary image
ret, bw_img = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# Display the binary image in Colab
cv2_imshow(bw_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Results: