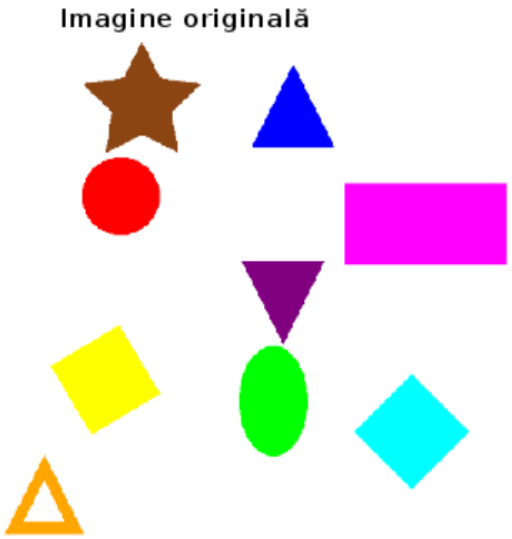
Cristea Dragos-Mihai
1306A

% % SHAPE DETECTION - Tema 2 %

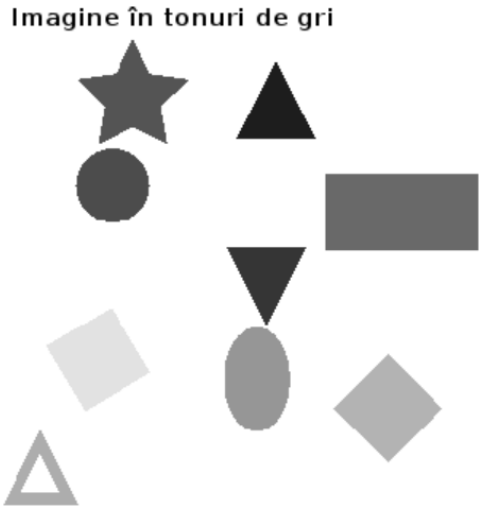
% Citirea imaginii

```
input_img = imread('1305B_1306A.png');
figure, imshow(input_img), title('Imagine originală');
```



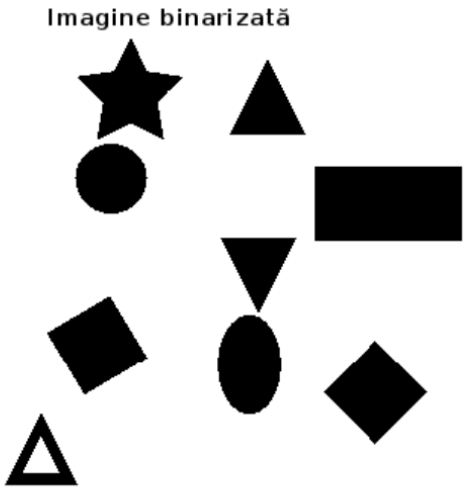
% Conversia imaginii RGB în tonuri de gri

```
gray_img = rgb2gray(input_img);
figure, imshow(gray_img), title('Imagine în tonuri de gri');
```



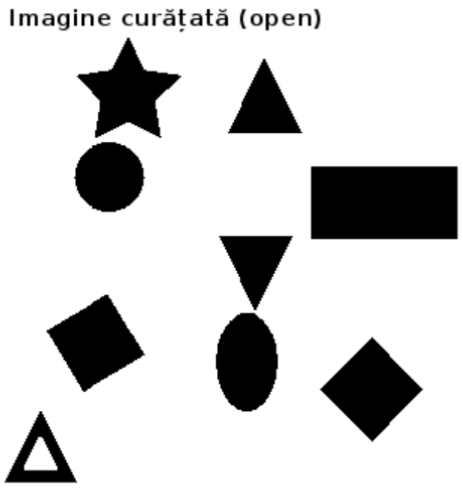
Cristea Dragos-Mihai
1306A

```
% Binarizarea imaginii folosind un prag fix
binary_img = imbinarize(gray_img, 0.9);
%prag de binarizare 0.9 pentru a face pixelii foarte luminosi albi
figure, imshow(binary_img), title('Imagine binarizată');
```

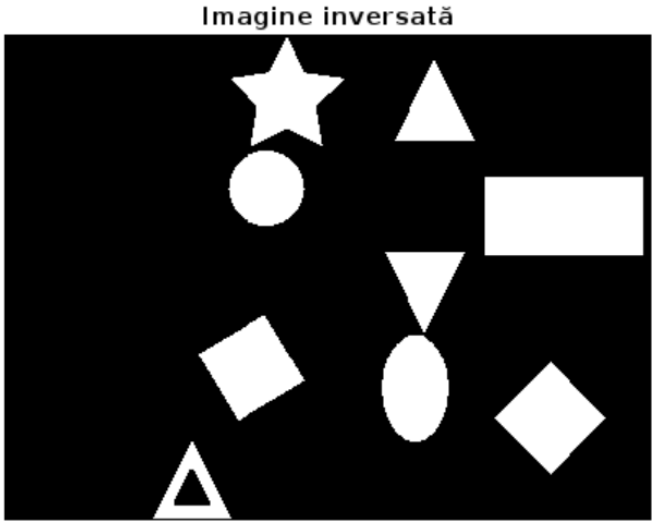



```
% Aplicarea operației morfologice de deschidere pentru a elimina zgomotul
struct_elem = strel('square', 5);           % Element structurant pătrat 5x5
cleaned_img = imopen(binary_img, struct_elem);
 %elimina zgomotul mic si subtire si reface forma obiectelor mari, pastrand contururile principale
figure, imshow(cleaned_img), title('Imagine curățată (open)');
```

Cristea Dragos-Mihai
1306A

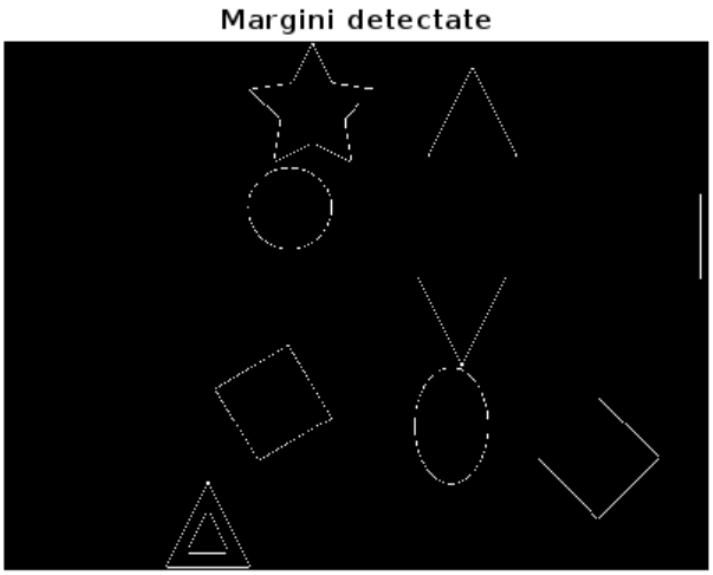



```
% Inversarea imaginii (pentru a obține obiectele albe pe fundal negru)
inverted_img = ~cleaned_img;
figure, imshow(inverted_img), title('Imagine inversată');
```




```
% Detectarea marginilor obiectelor din imagine
edge_img = edge(inverted_img);
figure, imshow(edge_img), title('Margini detectate');
```

Cristea Dragos-Mihai
1306A



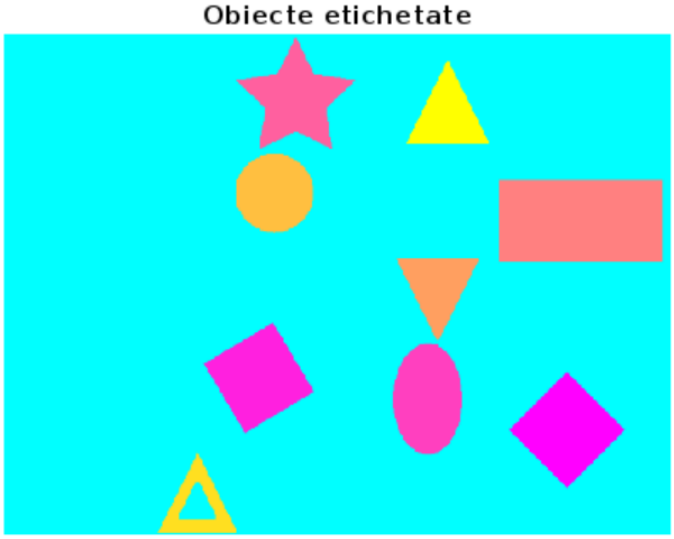
% Etichetarea componentelor conectate (obiecte)

[label_matrix, object_count] = bwlabel(inverted_img, 4);

% Colorarea etichetelor pentru vizualizare

colored_labels = label2rgb(label_matrix, @spring, 'c', 'shuffle');

figure, imshow(colored_labels), title('Obiecte etichetate');



% Calculul proprietăților pentru fiecare obiect detectat

object_stats = regionprops(logical(label_matrix), ...

```
    'Area', 'Perimeter', 'Eccentricity', 'Extent', 'BoundingBox', ...
    'Centroid', 'Solidity', 'Orientation');

% Inițializarea imaginii finale pentru adnotări
annotated_img = input_img;

% Parcurgerea tuturor obiectelor pentru a determina forma
for k = 1:object_count
    % Extragem proprietățile obiectului curent
    centroid_pos = object_stats(k).Centroid;
    obj_area = object_stats(k).Area;
    obj_perimeter = object_stats(k).Perimeter;
    compactness = obj_perimeter^2 / obj_area;      % metrică pentru formă
    obj_extent = object_stats(k).Extent;
    obj_ecc = object_stats(k).Eccentricity;
    obj_solidity = object_stats(k).Solidity;
    bbox = object_stats(k).BoundingBox;
    aspect_ratio = bbox(3) / bbox(4);
    orientation = object_stats(k).Orientation;

    % Metrică pentru circularitate
    circularity = 4 * pi * obj_area / (obj_perimeter^2);

    % Croim obiectul individual pentru a detecta colțuri
    sub_img = imcrop(inverted_img, bbox);
    corners_detected = corner(sub_img);
    corner_count = size(corners_detected, 1);

    % Inițializare formă
    detected_shape = 'Necunoscut';
```

Cristea Dragos-Mihai
1306A

```
    % Clasificarea formei pe baza proprietăților
    if obj_ecc < 0.2 && circularity > 0.85
        detected_shape = 'Cerc';
    elseif compactness > 19 && compactness <= 21
        if orientation < 0
            detected_shape = 'Triunghi întors';
        else
            detected_shape = 'Triunghi';
        end
    elseif obj_extent > 0.95 && aspect_ratio > 1
        detected_shape = 'Dreptunghi';
    elseif compactness > 15 && compactness <= 16
        if aspect_ratio >= 0.9 && aspect_ratio <= 1.1
            if orientation == 0
                detected_shape = 'Romb';
            else
                detected_shape = sprintf('Pătrat (%.1f°)', orientation);
            end
        else
            detected_shape = 'Dreptunghi';
        end
    elseif corner_count >= 15 && obj_solidity < 0.8
        detected_shape = 'Stea';
    elseif obj_solidity < 0.9 && obj_ecc < 0.8
        detected_shape = 'Triunghi gol';
    elseif obj_extent > 0.7 && obj_extent < 0.95 && obj_solidity > 0.85
        detected_shape = 'Elipsă';
    end

    % Inserarea formei recunoscute pe imagine
    annotated_img = insertText(annotated_img, centroid_pos, detected_shape, ...
```

Cristea Dragos-Mihai
1306A

```
        'BoxColor', 'white', 'TextColor', 'black', 'BoxOpacity', 0, 'FontSize', 20);


    % Trasarea dreptunghiului de delimitare în jurul obiectului
    annotated_img = insertShape(annotated_img, 'Rectangle', bbox, 'Color', 'black');
end


% Afișarea rezultatului final
figure, imshow(annotated_img), title('Forme detectate în imagine');
```