

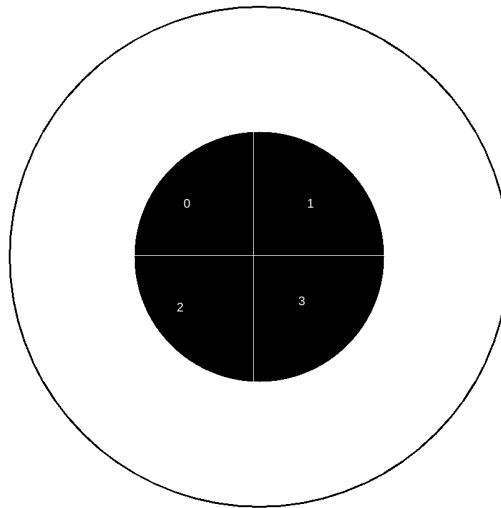
You have to design encoding and decoding algorithms based on binary values. The range of values to be converted is 0 to 15.

In the resource folder you can find box.png, dot.png, circle.png and some sample encoded output where the filename suggests the number encoded in image.

Encryption technique is simple, just the binary value. We have provided a dictionary for your reference inside label_encoder.py.

Problem 1:

In circle.png consider the solid circle to be divided into 4 equal pies and consider them 0,1,2,3. Refer to the image below-



Now, each pie should represent a binary value and position index refers to the value to be selected from the binary number. For example, binary value: (1000) will be assigned to the pie in following order-

Pie no 0 → 1

Pie no 1 → 0

Pie no 2 → 0

Pie no 3 → 0

Now a pie with binary value 1 should consist of dot.png inside the pie, with an offset (100 is preferred) from the center of the circle. Refer to sample output files in resources.

Problem 2:

Blend box.png and output from problem 1 to create the final output label and save the image with encoding number as name(as shown in sample output).

Problem 3:

Design a detection algorithm that reads the output image of problem 2, decodes the encryption and returns the decoded value in decimal. The detection should be dynamic in nature, no hardcoded pixel location should be used for decoding.

To start with, you have been provided with `label_encoder.py` and `label_decoder.py` with basic initiation. Preferred language to be used is python/C++.