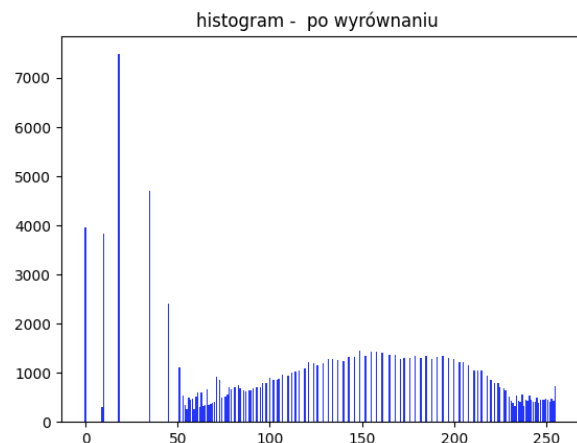
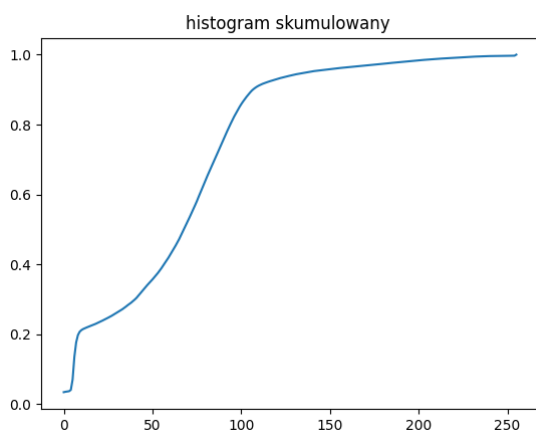
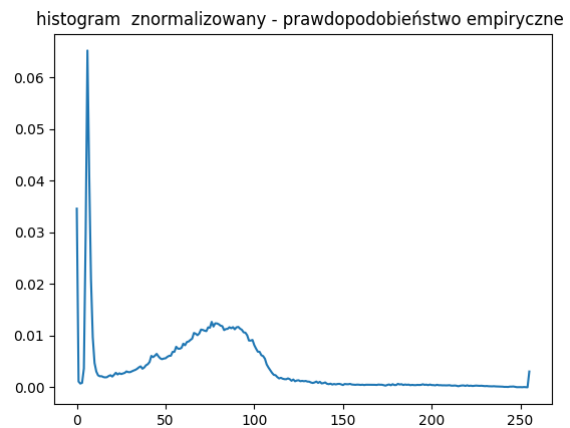
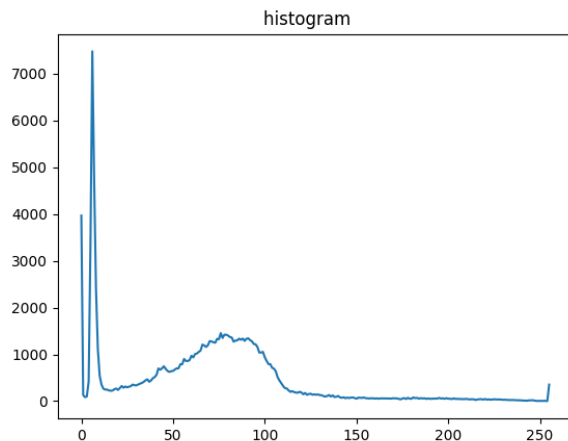


## LAB 9 Filtrowanie obrazu przez wyrównanie histogramu, konwersja na tryb 'L'

**Histogram w trybie 'L' :** każdej wartości z zakresu od 0 do 255 przyporządkowana jest liczba pikseli o tej wartości.



### Wyrównanie histogramu (dla obrazów w trybie 'L'):

1. Pobranie histogramu obrazu – lista `hist` długości 256
2. Normalizacja, tzn. każdy element histogramu dzielimy przez liczbę wszystkich pikseli w obrazie – lista `hist_norm` długości 256
3. Kumulacja, tzn. tworzymy z histogramu znormalizowanego histogram skumulowany – lista `hist_kumul` długości 256 – taka, że `hist_kumul[i]` jest sumą wszystkich elementów `hist_norm` o indeksach mniejszych równych `i`
4. Filtr obrazu przez wyrównanie histogramu, tzn. wartość `p` każdego piksela obrazu zamieniamy na `int(255*hist_kumul[p])`

## Zadania

1. Wczytaj obraz: `zeby.png`. Sprawdź tryb i przekonwertuj do trybu 'L'.

2. Napisz program

2.1 `histogram_norm(obraz)`, który na wyjściu daje histogram znormalizowany obrazu.

2.2 `histogram_cumul(obraz)`, który na wyjściu daje histogram skumulowany obrazu.

2.3 `histogram_equalization(obraz)`, który na wyjściu daje obraz powstały po wyrównaniu histogramu obrazu. Zastosuj tę funkcję do obrazu `obraz` i zapisz jako obraz `equalized.png`

2.4 przedstaw histogram obrazu oraz histogramy z punktów 2.1, 2.2, 2.3 na jednym diagramie plt (`fig1.png`)

2.5 skomentuj różnice między obrazem i obrazem otrzymanym po wyrównaniu histogramu na podstawie ich statystyk

3. Zastosuj metodę `ImageOps.equalize` do obrazu `obraz`, zapisz obraz jako `equalized1.png`. Następnie

3.1 przedstaw obraz wejściowy i obrazy z pkt. 2. i 3. na jednym diagramie plt (`fig2.png`)

3.2 skomentuj różnice między obrazami z pkt. 2. i 3. Dlaczego te obrazy są różne?

4. Napisz funkcję `konwertuj1(obraz, w_r, w_g, w_b)`, gdzie  $0 \leq w_r, w_g, w_b \leq 1$  oraz  $w_r + w_g + w_b = 1$ . Funkcja dokonuje konwersji obrazu w trybie RGB na tryb L, w ten sposób, że wartości pikseli liczone są jako suma ważona  $L = R \cdot w_r + G \cdot w_g + B \cdot w_b$ . Do zamiany liczb rzeczywistych na całkowite zastosuj `round`.

Zgodnie z dokumentacją `Image.convert('L')` działa następująco:

```
When translating a color image to greyscale (mode "L"),
the library uses the ITU-R 601-2 luma transform::
```

$$L = R * 299/1000 + G * 587/1000 + B * 114/1000$$

4.1 Zastosuj swoją funkcję do obrazu `mgla.jpg` stosując powyższe wagi i otrzymany obraz zapisz jako `mgla_L1.png` oraz zastosuj `Image.convert('L')` do obrazu `mgla.jpg` i zapisz jako `mgla_L.png`.

4.2 Oceń różnice między obrazami `mgla_L.png`, `mgla_L1.png` korzystając ze statystyk obrazów. Z jakiego powodu te obrazy mogą się różnić?

4.3 Powtórz punkty 4.1 (otrzymany obraz zapisz jako `mgla_L2.png`) i 4.2 stosując swoją funkcję `konwertuj2(obraz, w_r, w_g, w_b)`, w której zmienisz `round()` na `int()`. Czy różnice są większe? Uzasadnij odpowiedź.

Wstaw na Moodle: kod pythona, obrazy, raport