Raport 3-4

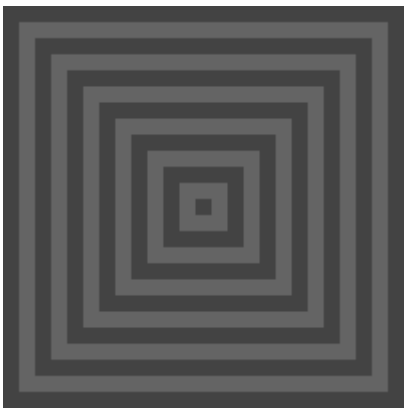
Wiktor Krasiński ISI gr. 2

Zad. 1.

```python
#------------------Zad 1------------------
def rysuj_ramki_szare(w, h, grub, kolor, kolor_ramki):  1 usage
    wymiary = (h, w)
    tab_obraz = np.ones(wymiary, dtype=np.uint8)
    ile = int(w/grub)

    for i in range(ile):
        top = i * grub
        right = w - i * grub
        bottom = h - i * grub
        left = i * grub

        if i % 2 == 1:
            tab_obraz[top:bottom, left:right] = kolor
        else:
            tab_obraz[top:bottom, left:right] = kolor_ramki

    return Image.fromarray(tab_obraz)

ramki_szare = rysuj_ramki_szare( w: 200,  h: 200,  grub: 8,  kolor: 100,  kolor_ramki: 67)
#ramki_szare.save("ramki_szare.png")
```

Po wykonaniu, otrzymujemy ramki_szare:

Reguła: ustalone kolory zmieniają się na przemian (kolor raz pierwszy a następnie drugi)



```python
def rysuj_pasy_pionowe_szare(w,h,grub, kolor):  4 usages
    t = (h, w)
    tab = np.ones(t, dtype=np.uint8)
    ile = int(w / grub)+1
    for k in range(ile):
        for g in range(grub):
            j = k * grub + g
            if j < w:
                for i in range(h):
                    tab[i, j] = (k + kolor) % 256
    return Image.fromarray(tab)

rysuj_pasy_pionowe_szare( w: 200,  h: 150,  grub: 10,  kolor: 30).show()
```

Po wykonaniu, otrzymujemy pasy_pionowe_szare:

Reguła: gradient, od lewej mamy ciemniejszy kolor który powoli jaśnieje do prawej

Zad. 2.

```
def negatyw(obraz):  2 usages
    if obraz.mode == "L":
        tab = np.asarray(obraz)
        h, w = tab.shape
        tab_neg = tab.copy()
        for i in range(h):
            for j in range(w):
                tab_neg[i, j] = 255 - tab[i, j]
        return Image.fromarray(tab_neg)

    elif obraz.mode == "1":
        tab = np.asarray(obraz)
        h, w = tab.shape
        tab_neg = tab.copy()
        for i in range(h):
            for j in range(w):
                tab_neg[i, j] = ~tab[i, j]
        return Image.fromarray(tab_neg)

    elif obraz.mode == "RGB":
        tab = np.asarray(obraz)
        h, w, _ = tab.shape
        tab_neg = tab.copy()
        for i in range(h):
            for j in range(w):
                tab_neg[i, j] = 255 - tab_neg[i, j]
        return Image.fromarray(tab_neg)
    return None
```

a) Po wykonaniu funkcji, otrzymujemy gwiazdka_negatyw (obrazek po prawej):


b) Po wykonaniu funkcji, otrzymujemy ramki_kolorowe_negatyw (obrazek po prawej):


c) Po wykonaniu funkcji, otrzymujemy po_skosie_szare_negatyw (obrazek po prawej):


Zad. 3.

```python
#------------------Zad 3------------------
def koloruj_w_paski(obraz, grub, kolor, kolor_paski):  1 usage
    if obraz.mode == "1":
        t = np.asarray(obraz)
        h, w = t.shape
        tab = np.ones( shape: (h, w, 3), dtype=np.uint8) * 255

        for i in range(h):
            k = i // grub
            r = (kolor[0] + k * kolor_paski) % 256
            g = (kolor[1] + k * kolor_paski) % 256
            b = (kolor[2] + k * kolor_paski) % 256
            for j in range(w):
                if t[i, j] == 0:
                    tab[i, j] = [r, g, b]
        return Image.fromarray(tab)
    return None
```

a) Po wykonaniu funkcji, otrzymujemy kolorowe_paski na inicjałach (obrazek po prawej):

b) Obrazy się różnią, obraz .jpg ma rozmazane kolory i ostrość, za to .png wygląda idealnie. Format JPG używa kompresji stratnej czyli kiedy zapisujemy obraz to algorytm usuwa część informacji o kolorach oraz ostrości, aby zmniejszyć rozmiar pliku co właśnie powoduje że wygląda on mniej czysto i ma plamy.

Zad. 4.

Zakres wartości uint8 to 0 - 255, jeśli nie damy zawijania (% 256) to numpy pokaże błąd o tym, że np. liczba 328 jest poza limitem uint8.

```
Traceback (most recent call last):  Explain with AI
  File "C:\Users\Wiktor\WprDoGrafMasz\lab4\lab4.py", line 173, in <module>
    r = np.asarray(rysuj_pasy_pionowe_szare(300, 200, 10 ,328))
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\Wiktor\WprDoGrafMasz\lab4\lab4.py", line 36, in rysuj_pasy_pionowe_szare
    tab[i, j] = (k + kolor)
    ~~~^^^^^^
OverflowError: Python integer 328 out of bounds for uint8
```

a)
328 % 256 = 72 (uint8)

b)
-24 % 256 = 232 (uint8)

Zad. 5.

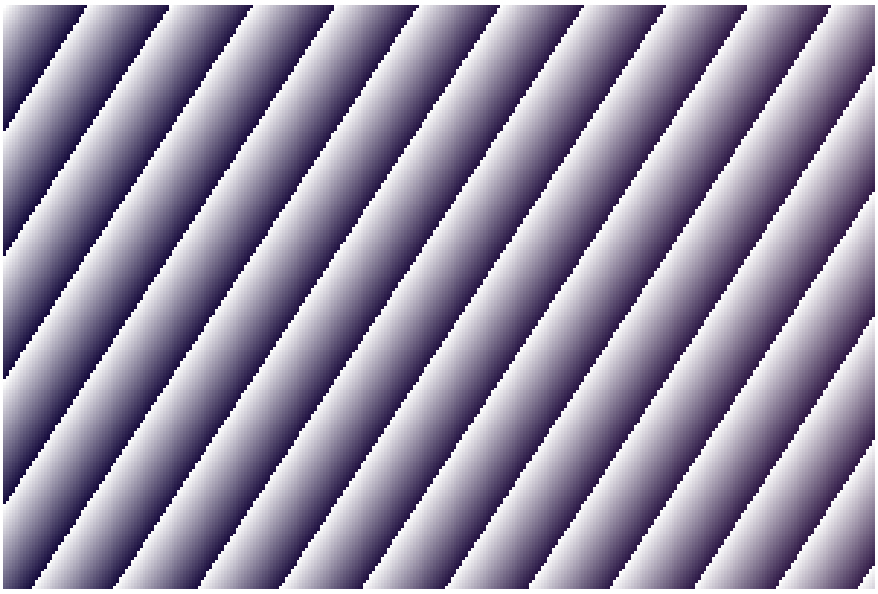
```
#------------------Zad 5------------------
r = np.asarray(rysuj_pasy_pionowe_szare( w: 300,  h: 200,  grub: 10 , kolor: 20))
g = np.asarray(rysuj_pasy_pionowe_szare( w: 300,  h: 200,  grub: 18 , kolor: 10))
b = np.asarray(rysuj_pasy_pionowe_szare( w: 300,  h: 200,  grub: 26 , kolor: 60))

rgb = np.stack( arrays: (r,g,b), axis=2)
obraz6 = Image.fromarray(rgb)
obraz6.show()
#obraz6.save("obraz6.png")
```

Po wykonaniu funkcji otrzymujemy obraz6:

Zad. 6.

```python
#------------------Zad 6------------------
def rysuj_po_skosie_szare(h,w, a, b):  # formuła zmiany wartości elemntów tablicy a*i + b*j  1 usage
    t = (h, w) # rysuje kwadratowy obraz
    tab = np.zeros(t, dtype=np.uint8)
    for i in range(h):
        for j in range(w):
            tab[i, j] = (a*i + b*j) % 256
    return tab


alfa = rysuj_po_skosie_szare( h: 200,  w: 300,  a: 6,  b: 9)
obraz_rgb = Image.open("obraz6.png")
alfa_ext = np.expand_dims(alfa, axis=-1)
combined = np.concatenate( arrays: (obraz_rgb, alfa_ext), axis=-1)

obraz7 = Image.fromarray(combined)
#obraz7.show()
#obraz7.save("obraz7.png")
```

Po wykonaniu funkcji otrzymujemy obraz7:



Zad. 7.

```python
#------------------Zad 7-----------------
def rgb_to_cmyk(rgb_array):  1 usage
    # Przekształć wartości RGB na zakres [0, 1]
    rgb = rgb_array.astype(float) / 255
    r, g, b = rgb[..., 0], rgb[..., 1], rgb[..., 2]

    # Oblicz kanał Kk (black)
    k = 1 - np.max(rgb, axis=2)

    # Uniknij dzielenia przez zero
    c = (1 - r - k) / (1 - k + 1e-8)
    m = (1 - g - k) / (1 - k + 1e-8)
    y = (1 - b - k) / (1 - k + 1e-8)

    # Zastąp NaN (dla czystej czerni) zerami
    c[np.isnan(c)] = 0
    m[np.isnan(m)] = 0
    y[np.isnan(y)] = 0

    # Przekształć na zakres [0, 255]
    cmyk = np.stack( arrays: (c, m, y, k), axis=2) * 255
    return cmyk.astype(np.uint8)


# Konwersja do CMYK
t_cmyk = rgb_to_cmyk(np.asarray(obraz6))

image_cmyk = Image.fromarray(t_cmyk, mode="CMYK")
#image_cmyk.save("obraz8.tiff")

# a)
r_png = Image.fromarray(r)
#r_png.save("r.png")

c_png = Image.fromarray(t_cmyk[:, :, 0])
#c_png.save("c.png")
```

Po wykonaniu funkcji otrzymujemy obraz8:

Kanał r.png:

Kanał c.png:

a) Kanał R ma regularną grubość pasków co tyle samo pikseli, a za to kanał C ma paski raz grubsze raz szersze co nieregularną ilość pikseli.

b) Sposobem na porównanie tych obrazów jest porównanie punktu po punkcie za pomocą średniej różnicy (tzw. Mean Absolute Error, inaczej MAE), lub innym sposobem jest sprawdzenie liniowej zależności między kanałami za pomocą korelacji (Pearson correlation).