Funkcja do tworzenia negatywu dla szarych obrazów:

```
def negatyw_szare(obraz):
    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 tab_neg
```

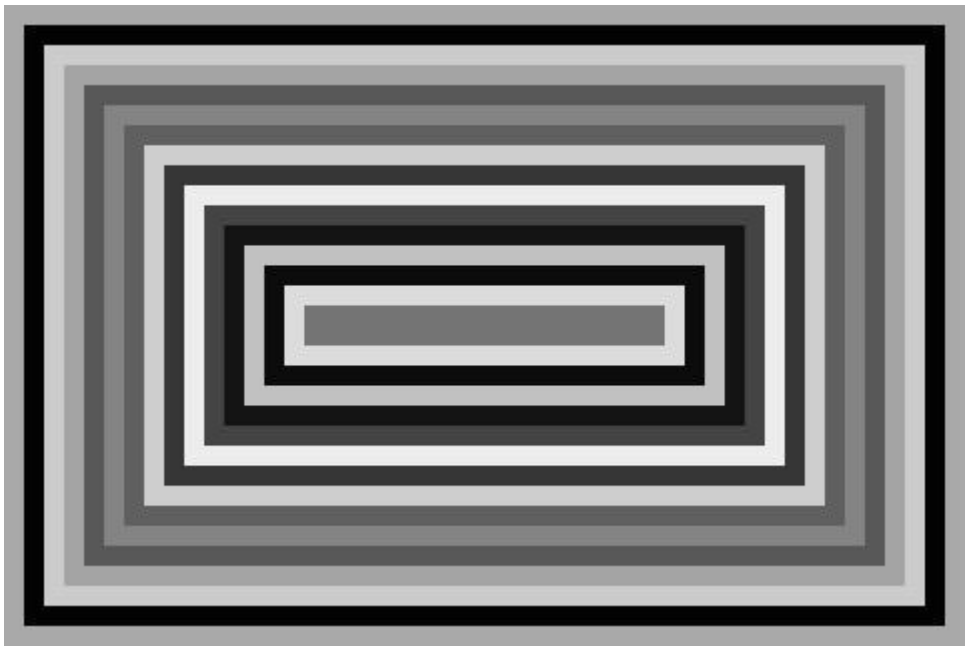
Do tworzenia kolorów dla pierwszych trzech zadań użyłem metody `randint(0,255)` z biblioteki random: `import random`

Zadanie 1:

Dla zadań 1-2 użyłem funkcji z lab2 które generowały ramke w ramce oraz pionowe paski które przerobiłem tak aby generowały kolory szarości

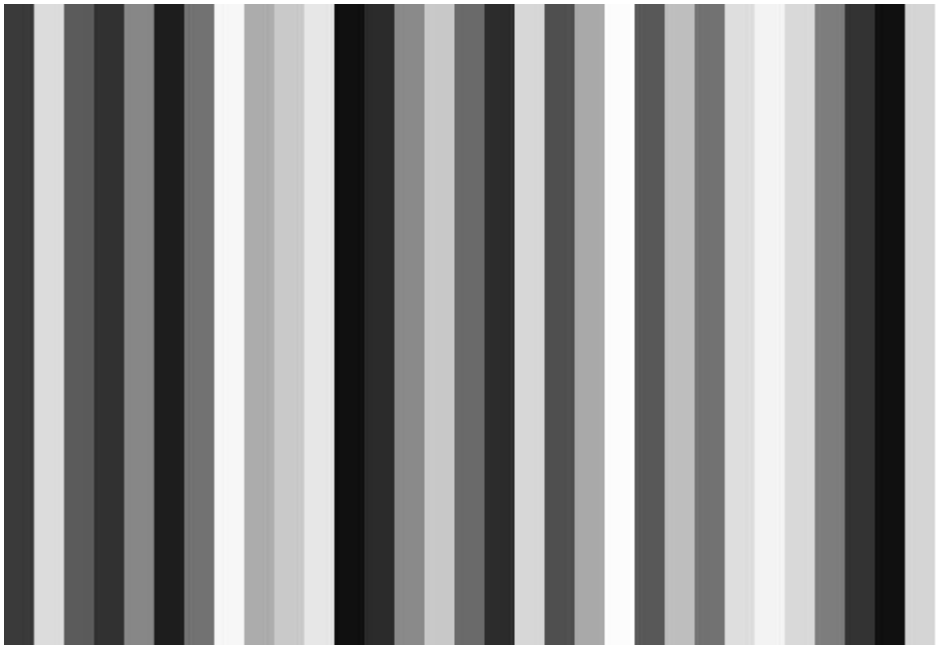
```
def rysuj_ramke(w, h, grub):
    t = (h, w)
    tab = np.ones(t, dtype=np.uint8)
    for j in range(int(min(w,h)/grub)):
        tab[j*grub:h - j*grub, j*grub:w - j*grub] = random.randint(0,240)
    return tab

ramka = Image.fromarray(rysuj_ramke(480, 320, 10))
ramka.save("obraz1_1.jpg")
ramka.save("obraz1_1.png")
ramka2 = Image.fromarray(negatyw_szare(ramka))
ramka2.save("obraz1_1N.jpg")
ramka2.save("obraz1_1N.png")
```

Wynik dla `ramka.save("obraz1_1.jpg"`



```
def pionowe_paski(w, h, grub):
    t = (h, w)
    tab = np.ones(t, dtype=np.uint8)
    ile2 = int(w / grub)
    j = 0
    tab_pom = [10,20,30,50]
    for k in range(ile2):
        tab[:,j:j+grub] = random.randint(0,250)
        j = k * grub
    tab = tab * 255
    return Image.fromarray(tab)

pionowe = pionowe_paski(480, 320, 15)
pionowe.save("obraz1_2.jpg")
pionowe.save("obraz1_2.png")
pionowe_neg = Image.fromarray(negatyw_szare(pionowe))
pionowe_neg.save("obraz1_2N.jpg")
pionowe_neg.save("obraz1_2N.png")
```

Wynik dla `pionowe.save("obraz1_2.jpg")`

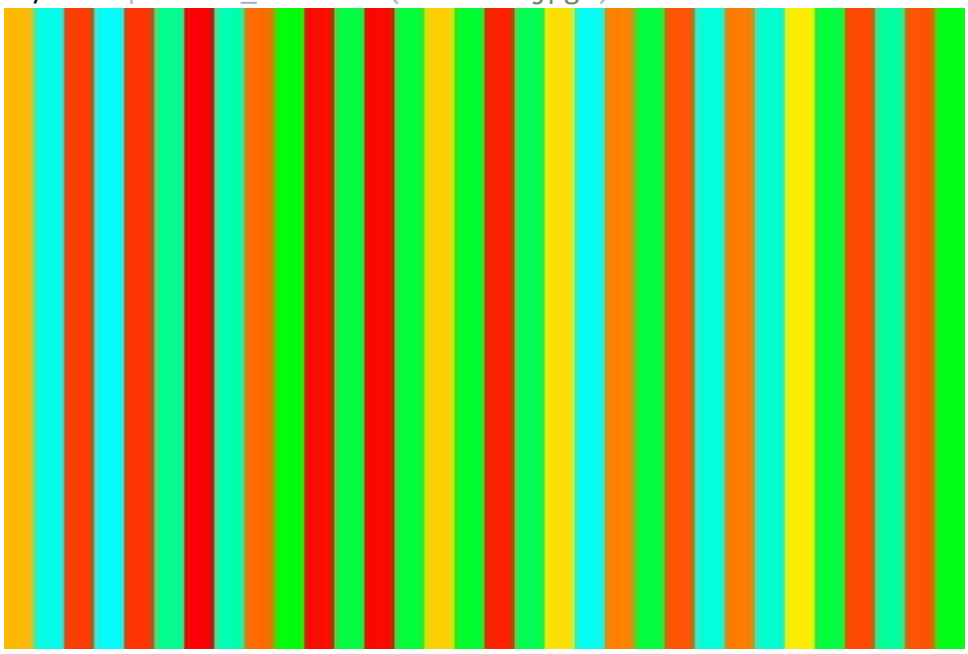


Zadanie 2:

Funkcja do tworzenia negatywu dla obrazów RGB:

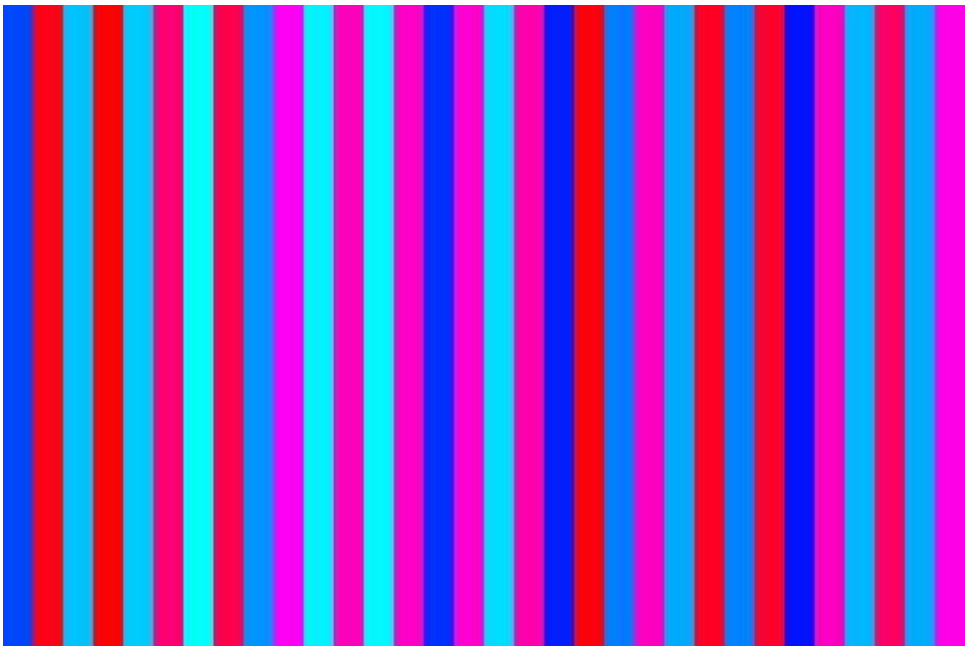
```
def negatyw_RGB(obraz):
    tab = np.asarray(obraz)
    h, w, c = tab.shape
    tab_neg = tab.copy()
    for i in range(h):
        for j in range(w):
            for k in range(c):
                tab_neg[i, j, k] = 255 - tab[i, j, k]
    return Image.fromarray(tab_neg)


def pionowe_paski_RGB(w, h, grub):
    t = (h, w, 3)
    tab = np.zeros(t, dtype=np.uint8)
    ile2 = int(w / grub)
    j = 0
    for k in range(ile2+1):
        if k%2 == 0:
            tab[:,j:j+grub] = [0,255,random.randint(0,255)]
        else:
            tab[:, j:j + grub] = [255, random.randint(0,255), 0]
        j = k * grub
    return Image.fromarray(tab)

pionowe_RGB = pionowe_paski_RGB(480, 320, 15)
test = np.asarray(pionowe_RGB)
pionowe_RGB.save("obraz2.jpg")
pionowe_RGB.save("obraz2.png")
```

Wynik dla `pionowe_RGB.save("obraz2.jpg")`



Wynik dla:

```
pionowe_RGB_negatyw = negatyw_RGB(pionowe_RGB)
pionowe_RGB_negatyw.save("obraz2N.jpg")
```



Zadanie 3:

```
def koloruj_obraz(obraz, grubosc_paskow):
    t_obraz = np.asarray(obraz)
    h, w = t_obraz.shape
    t = (h, w, 3)
    tab = np.ones(t, dtype=np.uint8)
    for i in range(h):
        for j in range(w):
            if t_obraz[i, j] == False:
                if (i // grubosc_paskow) % 2 == 0:
                    tab[i, j] = [66, 222, 23]
                else:
                    tab[i, j] = [66, 11, 222]
            else:
                tab[i, j] = [255, 255, 255]
    return tab

#gwiazdka = Image.open("gwiazdka.bmp")
obraz2 = Image.fromarray(koloruj_obraz(inicjaly, 15))
obraz2.show()
obraz2.save("obraz3.jpg")
obraz2.save("obraz3.png")
```

Zadanie 5:

Różnice między JPG a PNG widać po większym przybliżeniu obrazka gdzie w formacie PNG krawędzie oraz kolory są ostre ale w formacie JPG przy krawędziach jest cieniowanie oraz widać "rozpikselowane" krawędzie