

```
# Zad.1
2 usages
def rysuj_ramke_w_obrazie(obraz, grub):
    tab_obraz = np.asarray(obraz)*1
    h, w = tab_obraz.shape
    for i in range(h):
        for j in range(grub):
            tab_obraz[i][j] = 0
        for j in range(w-grub, w):
            tab_obraz[i][j] = 0
    for i in range(w):
        for j in range(grub):
            tab_obraz[j][i] = 0
        for j in range(h-grub, h):
            tab_obraz[j][i] = 0
    tab = tab_obraz.astype(bool)
    return Image.fromarray(tab)
```

```
# Zad.2
inicjaly = Image.open("inicjaly.bmp")

ramka_10 = rysuj_ramke_w_obrazie(inicjaly, grub: 10)
ramka_10.save("ramka5.bmp")
ramka_5 = rysuj_ramke_w_obrazie(inicjaly, grub: 5)
ramka_5.save("ramka10.bmp")
```



```
# Zad.3_1
1 usage
def rysuj_ramke_w_ramce(w, h, grub):
    t = (h, w)
    tab = np.zeros(t, dtype=np.uint8)
    temp = 1
    minimum = min(w, h)
    end = int(minimum/(2*grub))
    for i in range(0, end):
        tab[temp*grub: h - (temp * grub), temp*grub: w - (temp*grub)] = temp % 2
        temp += 1
    tab2 = tab.astype(bool)
    return Image.fromarray(tab2)
```

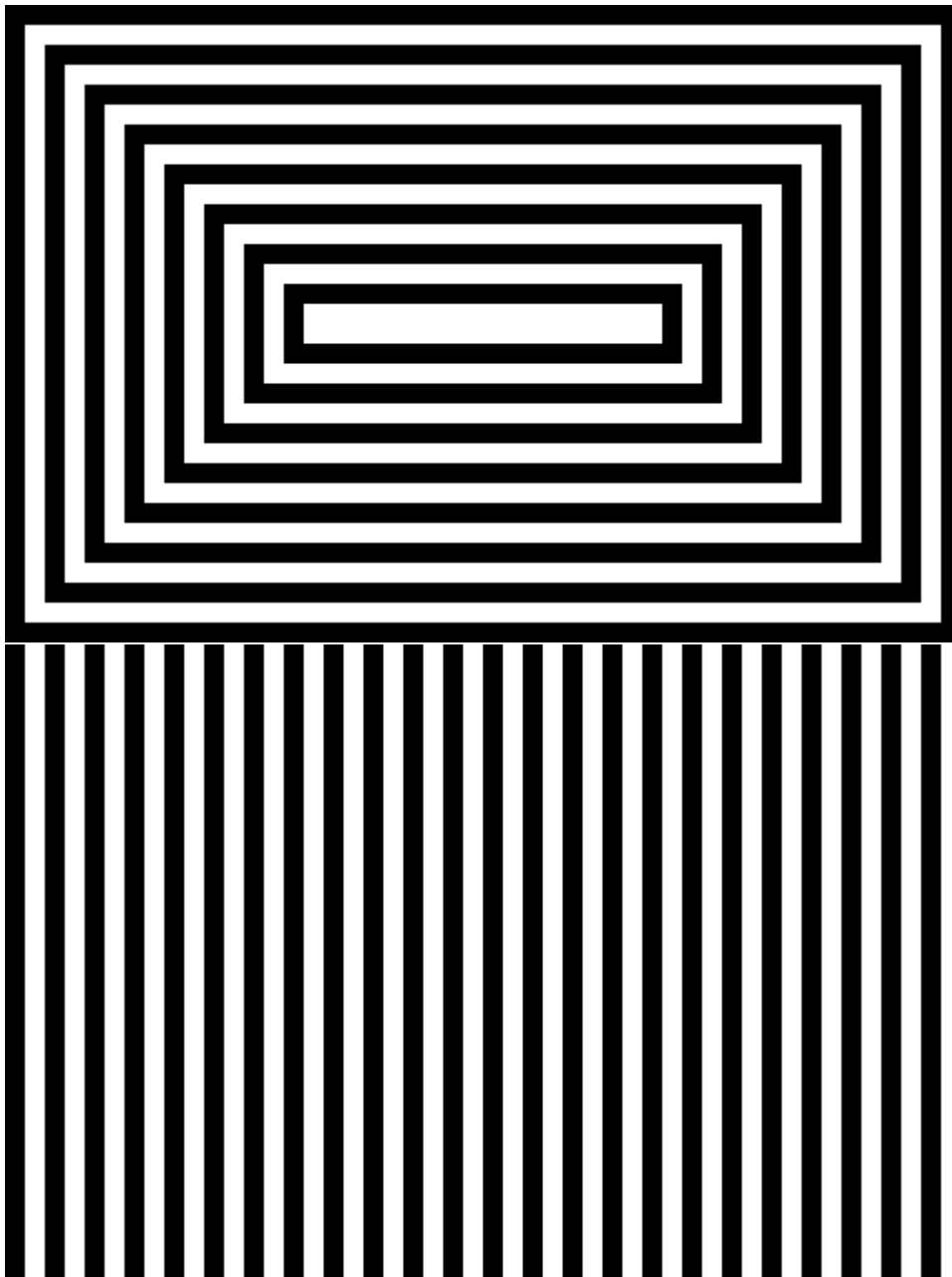
```
# Zad.3_2
1 usage
def rysuj_pasy_pionowe(w, h, grub):
    t = (h, w)
    tab = np.ones(t, dtype=np.uint8)
    ile = int(w/grub)
    for k in range(ile):
        for g in range(grub):
            i = k * grub + g
            for j in range(h):
                tab[j, i] = k % 2
    tab2 = tab.astype(bool)
    return Image.fromarray(tab2)
```

```
# Zad.3_3
1 usage
def styk_prostokatow(w, h, m, n):
    t = (h, w)
    tab = np.ones(t, dtype=np.uint8)
    tab[0:n, 0:m] = 0
    tab[n:h, m:w] = 0
    tab2 = tab.astype(bool)
    return Image.fromarray(tab2)
```

Cieniowanie od góry do dołu o ustawioną wartość (grubość).

```
# Zad.3_4
1 usage
def cieniowanie_od_gory(w, h, grubosc):
    wielkosc = (h, w)
    tab = np.ones(wielkosc, dtype=np.uint8)
    for i in range(w):
        for j in range(grubosc):
            tab[j][i] = 0
    tab2 = tab.astype(bool)
    return Image.fromarray(tab2)
```

```
# Zad.4
obraz_1 = rysuj_ramke_w_ramce(w: 480, h: 320, grub: 10)
obraz_1.save("obraz_1.bmp")
obraz_2 = rysuj_pasy_pionowe(w: 480, h: 320, grub: 10)
obraz_2.save("obraz_2.bmp")
obraz_3 = styk_prostokatow(w: 480, h: 320, m: 10, n: 50)
obraz_3.save("obraz_3.bmp")
obraz_4 = cieniowanie_od_gory(w: 480, h: 320, grubosc: 150)
obraz_4.save("obraz_4.bmp")
```



1

[REDACTED]

[REDACTED]

```
# Zad.5_1
2 usages
def wstaw_obraz_w_obraz(obraz_bazowy, obraz_wstawiany, m, n):
    tab_wstawiany = np.asarray(obraz_wstawiany) * 1
    hw, ww = tab_wstawiany.shape
    tab_bazowy = np.asarray(obraz_bazowy) * 1
    hb, wb = tab_bazowy.shape
    n_k = min(hb, n + hw)
    m_k = min(wb, m + ww)
    n_p = max(0, n)
    m_p = max(0, m)
    print(n_k, m_k)
    print(n_p, m_p)
    for i in range(n_p, n_k):
        for j in range(m_p, m_k):
            tab_bazowy[i][j] = tab_wstawiany[i - n][j - m]
    tab = tab_bazowy.astype(bool)
    return Image.fromarray(tab)
```

```
# Zad.5_2
wstaw = Image.open("obraz_2.bmp")
inicjaly = Image.open("inicjaly.bmp")
wstaw_1 = wstaw_obraz_w_obraz(wstaw, inicjaly, m: 300, n: 90)
wstaw_1.save("wstaw1.bmp")
wstaw_2 = wstaw_obraz_w_obraz(wstaw, inicjaly, m: 10, n: 290)
wstaw_2.save("wstaw2.bmp")
```

**MP**

**MD**