**Zadanie 1**

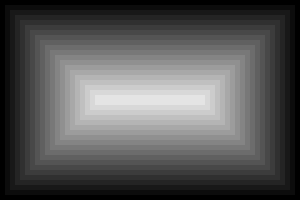
- **rysuj\_ramki\_szare**: Kolor ramek ciemnieje stopniowo, dzieląc 255 przez liczbę ramek.

Kod:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Rezultat:



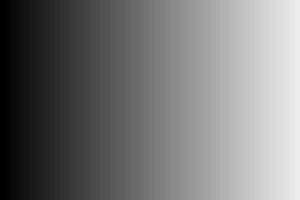
- **rysuj\_pasy\_pionowe\_szare**: Kolor pasów ciemnieje równomiernie, dzieląc 255 przez liczbę pasów.

Kod:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Rezultat:



**Zadanie 2**

Kod negatyw:

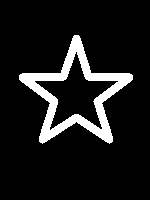
Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

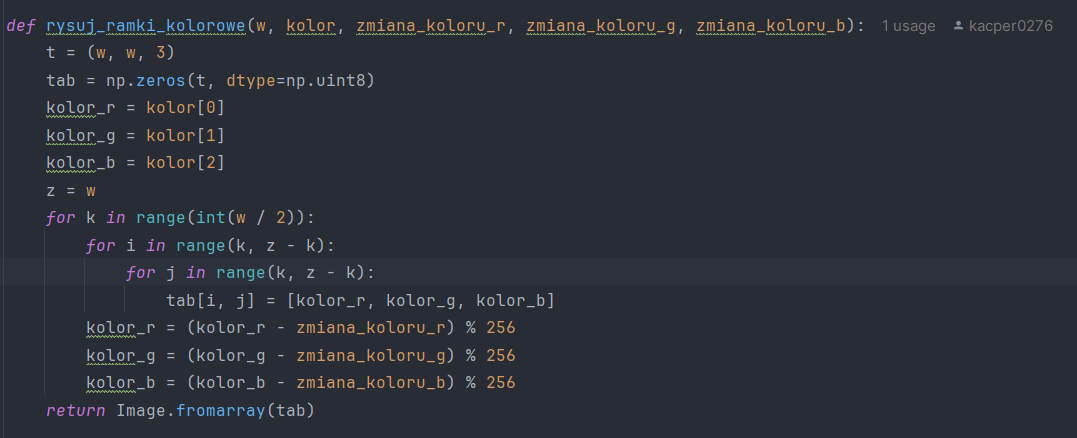
Gwiazdka oryginalna



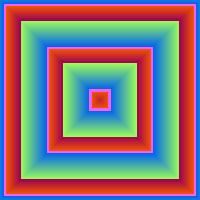
Gwiazdka negatyw



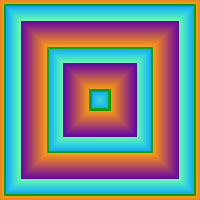
Ramki kolorowe kod:



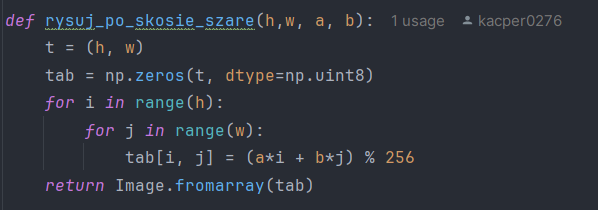
Ramki kolorowe oryginalne



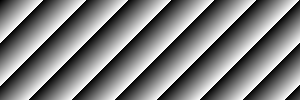
Ramki kolorowe negatyw



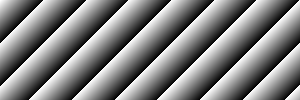
Po skosie kod:



Po skosie



Po skosie negatyw



**Zadanie 3**

Parametry:

- kolor: Początkowy kolor dla pierwszego paska, podany jako trzy wartości RGB

- zmiana\_koloru\_r: Zmiana wartości kanału czerwonego (R) po każdej zmianie pasków.

- zmiana\_koloru\_g: Zmiana wartości kanału zielonego (G) po każdej zmianie pasków.

- zmiana\_koloru\_b: Zmiana wartości kanału niebieskiego (B) po każdej zmianie pasków.

Kod:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

JPG

****

PNG



Otrzymane kolory nie są takie same, ponieważ jpg używa kompresji stratnej, co powoduje utratę jakości obrazu, natomiast png stosuje kompresję bezstratną, co oznacza, że zachowuje wszystkie szczegóły obrazu.

Dodatkowo png ma większą głębię kolorów

**Zadanie 4**

Kod służący do testów



Maksymalna wartość uint8 to 255 a minimalna 0, jest tak ponieważ 11111111 co po zmianie na zapis dziesiętny daje nam 255 taka liczba binarna jest maksymalna w tym typie.

Numpy w przypadku, gdy wartość jest większa niż 255 lub mniejsza niż 0 pojawia błąd **OverflowError: Python integer 328 out of bounds for uint8.** Więc jest tutaj walidacja tego.

W Internecie natknąłem się na taką metodę gdy wartości wykraczają poza zakres

**def uint8\_wrap(value):   
 if value < 0:   
 return value + 256  
 return value % 256**

Dla wartości -24 po zastosowaniu tej funkcji wychodzi liczba 232,

natomiast dla 328 jest ona równa 72