

# Rysowanie linii

---

Ostatnie zadanie w tym semestrze będzie wymagało niewiele pisania kodu, natomiast trzeba będzie poznać algorytm rysowania linii, konkretnie algorytm Bresenhama.

Jego dość dobry opis można znaleźć w Wikipedii:

[https://en.wikipedia.org/wiki/Bresenham%27s\\_line\\_algorithm](https://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm)

Warto poświęcić chwilę na jego zrozumienie, bo z implementacją nie powinno być problemu.

Dla potrzeb rysowania linii (i ew. innych operacji graficznych) warto mieć zdefiniowaną strukturę punktu:

```
typedef struct
{
    unsigned int x, y;
} Point;
```

# Rysowanie linii

---

Mając taką strukturę możemy zdefiniować nagłówek funkcji rysującej linię:

```
void bline(ImageInfo* pImg, Point start,  
           Point stop, unsigned char color);
```

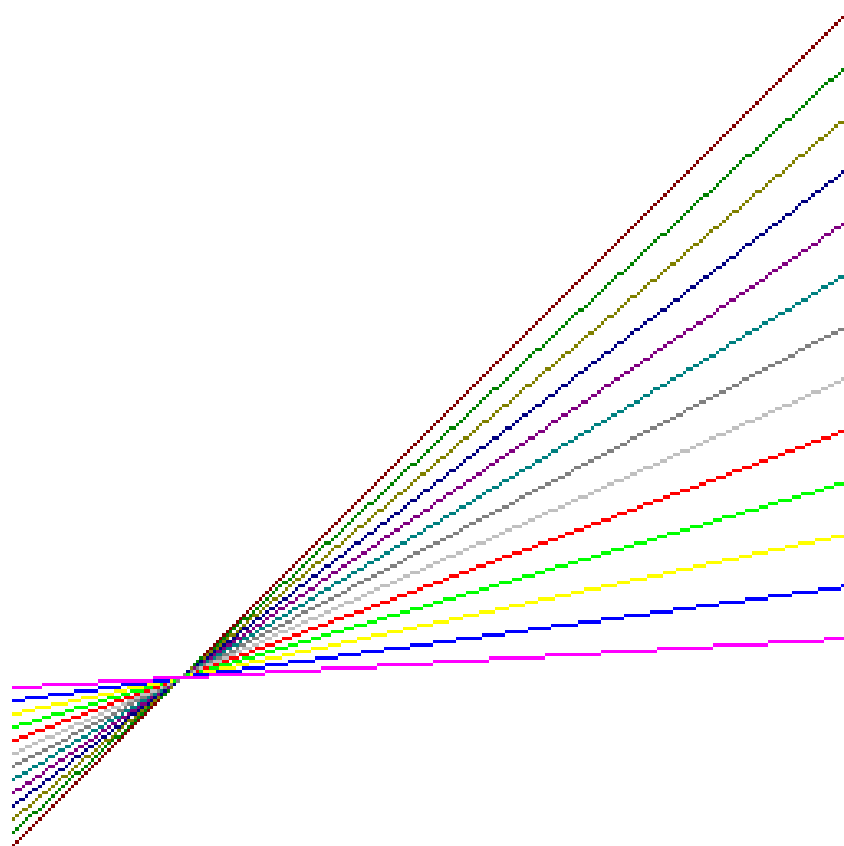
Warto pamiętać, że można zakładać, że współrzędne punktu start mają wartości mniejsze niż współrzędne punktu stop – to oznaczałoby, że jesteśmy ograniczeni tylko do jednej ćwiartki.

W ostatnim zadaniu możemy jednak przyjąć, że współrzędne będą mieściły się w obszarze obrazu.

# Wynik

---

Zachęcam do pokazania niestandardowych obrazków złożonych z odcinków prostych. Moja wersja wygląda tak:



# Wynik

---

Uzyskałem to stosunkowo prostym kodem (jest w zad\_x.c), którego główna pętla wygląda tak:

```
start.x = start.y = 0;
stop.x = stop.y = 255;
unsigned char color = 1;
while (start.y < stop.y)
{
    bline(pImg, start, stop, color);
    start.y += 4;
    stop.y -= 16;
    color++;
}
```

# Oddawanie

---

Rozwiązanie zadania składa się z 3 plików:

- `image.h` – zawiera deklaracje struktur i funkcji
- `image.c` – implementacje funkcji (bline jest istotne)
- `zad_x.c` – funkcja `main` zawierająca (być może) niestandardowy obraz.

Rozwiązanie proszę załadować w Moodle do:

**11 czerwca 23:59**