# DOKUMENTACJA PROJEKTU

**Politechnika Krakowska**
**Wydział inżynierii Elektrycznej i Komputerowej**

**Grafika Komputerowa i Multimedia**
Informatyka, grupa 22i

**Wykonali:**
Dawid Przystasz
Paweł Paszkowski
Tomasz Krzywonos
Bartłomiej Kapusta

1. **Temat:**
    1. Część:
        - Pierwszy program: zapis wartości pixeli z obrazka w formacie BMP (24 bit) do pliku binarnego, 5 bitów na kanał.
        - Drugi program: odczyt z wygenerowanego pliku binarnego i eksport BMP.

    2. Część:
        - Pierwszy program: Kodowanie Arytmetyczne lub Byterun na 5-bitowych wartościach pobranych z obrazka w formacie BMP (24 bit).
        - Drugi program: odczyt z wygenerowanego pliku binarnego, dekodowanie i eksport BMP.

    Obie części projektu są zaimplementowane w załączonych programach.

2. **Działanie:**
    1. Dla programu **btf** (**B**MP **t**o **F**ile) podajemy argumenty wiersza poleceń:
        - Jako pierwszy argument ścieżkę do pliku BMP (z nazwą i rozszerzeniem .bmp)
        - Jako drugi argument 0-2 dla rodzaju kodowania:
            - 0 – kodowanie arytmetyczne (na wartościach 5-bitowych)
            - 1 – kodowanie Byterun (na wartościach 5-bitowych)
            - 2 – skalowanie wartości na 5 bitowe i zapis jedna obok drugiej
        - Jako trzeci argument 0-1 dla opcji skala szarości:
            - 0 – z zachowaniem kolorów
            - 1 – przejście na odcienie szarości

        Program ten pobiera wartości składowych kolorów pikseli z pliku BMP i skaluje je z proporcji: **fiveBitValue = eightBitValue*31/255** na wartość mieszczącą się na 5 bitach. Następnie w zależności od wybranej metody (co zapisuje w nagłówku) albo stosuje któryś z algorytmów kompresji (cz.2), albo zapisuje te wartości jedna obok drugiej w pliku (cz.1).

    2. Dla programu **ftb** (**F**ile **t**o **B**MP) podajemy:
        - Jako jedyny argument ścieżkę do pliku binarnego (z nazwą i rozszerzeniem .file)

        Program ten odczytuje nagłówek pliku binarnego i odpowiednio pobiera wartości (liczby zmiennoprzecinkowe, całkowite jedno-bajtowe, bądź serie liczb 5 bitowych) z pliku. Następnie dokonuje odkodowania wartości składowych kolorów pikseli, skaluje je z powrotem z proporcji:
        **eightBitValue = fiveBitValue*255/31**,
        po czym zapisuje je i eksportuje plik BMP.

3. **Nagłówek:**
   - Rozmiar: 72 bitów (9B)
   - 16 bitów na szerokość obrazka
   - 16 bitów na wysokość obrazka
   - 32 bity na ilość wartości w pliku (wartości double dla kodowanie arytmetycznego, wartości char dla kodowania Byterun albo piątek bitów dla skalowania)
   - 1 bajt na: opcję skali szarości (1 bit: 0 zachowanie kolorów, 1 przejście na odcienie szarości) na MSB bajtu (5), numer metody na pozostałych jego bitach (0 kodowanie arytmetyczne, 1 Byterun, 2 skalowanie na wartości 5-bitowe)

   Przykładowy nagłówek:* (wygląd w pamięci dla Little Endian)

```
  (2)      (1)       (4)      (3)       (8)      (7)      (6)      (5)       (9)
11101010 00000000  10001010 00000000  00100100 01111110 00000000 00000000  10000010
[ szer. obrazka ]  [  wys. obrazka ]  [      ilość wartości w pliku     ]  [metoda]
```

   W tym przykładzie:
   - Obrazek 234px (szer.) x 138px (wys.)
   - 32292 wartości w pliku
   - Metoda: skalowanie na 5 bitów
   - Przejście na odcienie szarości

   *Numery w nawiasach to kolejność bajtów, adresy rosną w prawo. Kolejność wynika z tego, że zapisujemy header do pliku najpierw po 2 bajty na raz, (1) i (2), potem (3) i (4), następnie 4 bajty (5-8), a na końcu 1 bajt (9).

4. **Dalsza zawartość pliku:**
   Dalej plik binarny zawiera odpowiednio:
   - W przypadku kodowania arytmetycznego serię wartości typu **double** (po 64 bity)
   - W przypadku kodowania Byterun serię wartości typu **char** (po 8 bitów). (Wartości, chociaż mieszczą się na 5 bitach są tutaj nadal zapisane na ośmiu.)
   - W przypadku skalowania na 5 bitów serię 40-bitowych bloków

5. **Szczegóły dotyczące algorytmów kompresji:**
   - Algorytm Byterun dla ustawionej opcji skali szarości koduje powtarzające się pojedyncze wartości składowych kolorów, natomiast gdy opcja skali szarości nie jest ustawiona taka kompresja nie sprawdza się. Dlatego w takim przypadku algorytm koduje powtarzające się trójki wartości składowych (całe kolory), co sprawdza się, zwłaszcza gdy obrazek zawiera obszary o podobnych kolorach. Podobnych, a niekoniecznie identycznych na oryginalnym obrazku, z tego względu, że zawsze odbywa się skalowanie na wartości mieszczące się na 5 bitach, co powoduje że zwiększają się rozbieżności pomiędzy podobnymi kolorami i powstaje więcej identycznych pikseli obok siebie.

   - W algorytmie kodowania arytmetycznego na każdą liczbę zmiennoprzecinkową (podwójnej precyzji, 64 bity) przypada 10 wartości całkowitych.

6. **Jakość kompresji:**
   Poniżej podane zostały rozdzielczości, rozmiary i procentowy stosunek rozmiarów wejściowy plik BMP/wynikowy plik binarny. Testowane obrazki: (24-bit BMP)

- Losowe kolory pixeli, 256x256 px, 196.7 kB:
  - Kodowanie arytmetyczne: 157.3 kB (125%) ; 52.4kB (375%) (grayscale)
  - Byterun: 197.0 kB (99%) ; 68.4 kB (288%) (grayscale)
  - Skalowanie na 5 bitów: 122.9 kB (160%) ; 41.0 kB (480%) (grayscale)

- Wszystkie kolory RGB na jednym obrazku, 2048x1024 px, 6.3 MB:
  - Kodowanie arytmetyczne: 5.0 MB (126%) ; 1.7 MB (371%) (grayscale)
  - Byterun: 6.2 MB (101%) ; 2.1 MB (300%) (grayscale)
  - Skalowanie na 5 bitów: 3.9 MB (162%) ; 1.3 MB (485%) (grayscale)

- Zdjęcie krajobrazu w dobrej jakości, 3872x2592 px, 30.1 MB:
  - Kodowanie arytmetyczne: 24.1 (124%) ; 8.0 MB (376%) (grayscale)
  - Byterun: 15.7 MB (192%) ; 4.8 MB (627%) (grayscale)
  - Skalowanie na 5 bitów: 18.8 MB (160%) ; 6.3 MB (478%) (grayscale)

- Obrazek w całości w jednym kolorze , 640x400 px, 768.1 kB:
  - Kodowanie arytmetyczne: 614.4 kB (125%) ; 204.8 kB (375%) (grayscale)
  - Byterun: 8.0 kB (9601%) ; 4.0 kB (19202%) (grayscale)
  - Skalowanie na 5 bitów: 480.0 kB (160%) ; 160 kB (480%) (grayscale)