

Grafika Komputerowa i Multimedia

Wykład 12

Stratna kompresja obrazów

Damian Grela

e-mail: dgrela@pk.edu.pl

<http://www.dgrela.pl>



```
static public String show...
String str = Integer.toString(1234);
int count = leadingZeros(count - str.length());
```

```
begin
variable X : int;
begin
X := 2 and 0;
after 10;
end
```

- **Stratna kompresja obrazów:**
 - Algorytm JPEG
 - Algorytm JPEG2000



- Algorytmy predykcyjne
 - na podstawie
 - modelu obrazu
 - już przetworzonej części obrazu
 - znanego otoczenia danego piksela
 - spróbuj przewidzieć barwę piksela (predykcja)
 - kompresuj błąd predykcji (różnicę między przewidzianą i rzeczywistą barwą)

- Algorytmy transformacyjne, kodowanie podpasmowe
 - obraz przekształć odpowiednią transformacją (DCT, DWT)
 - kodowania predykcyjnego nie uznaje się za kodowanie transformacyjne
 - kompresuj macierz współczynników transformaty
 - stosowane głównie w algorytmach stratnych

- Opracowany przez Komitet JPEG w 1992 r.
- Popularny do dziś transformacyjny algorytm kompresji stratnej obrazów barwnych oraz w stopniach szarości, oparty o całkowitoliczbową transformatę kosinusową DCT oraz koder Huffmana
 - algorytm bazowy (baseline codec)
 - rozszerzenia (extensions)
 - Wallace, G. K.: The JPEG Still Picture Compression Standard. Communications of the ACM, April 1991, Vol 34(4), s. 30-44.

1. JPEG jest formatem przeznaczonym do zapisu obrazów nieruchomych z zastosowaniem kompresji stratnej.
2. JPEG obsługuje obrazy w odcieniach szarości (8 bitów na piksel) i RGB (24 bity na piksel).
3. JPEG wykorzystuje dwie cechy ludzkiego wzroku: mniejszą rozdzielczość widzenia kolorów niż jasności oraz małą wrażliwość na amplitudę szybkich zmian jasności.

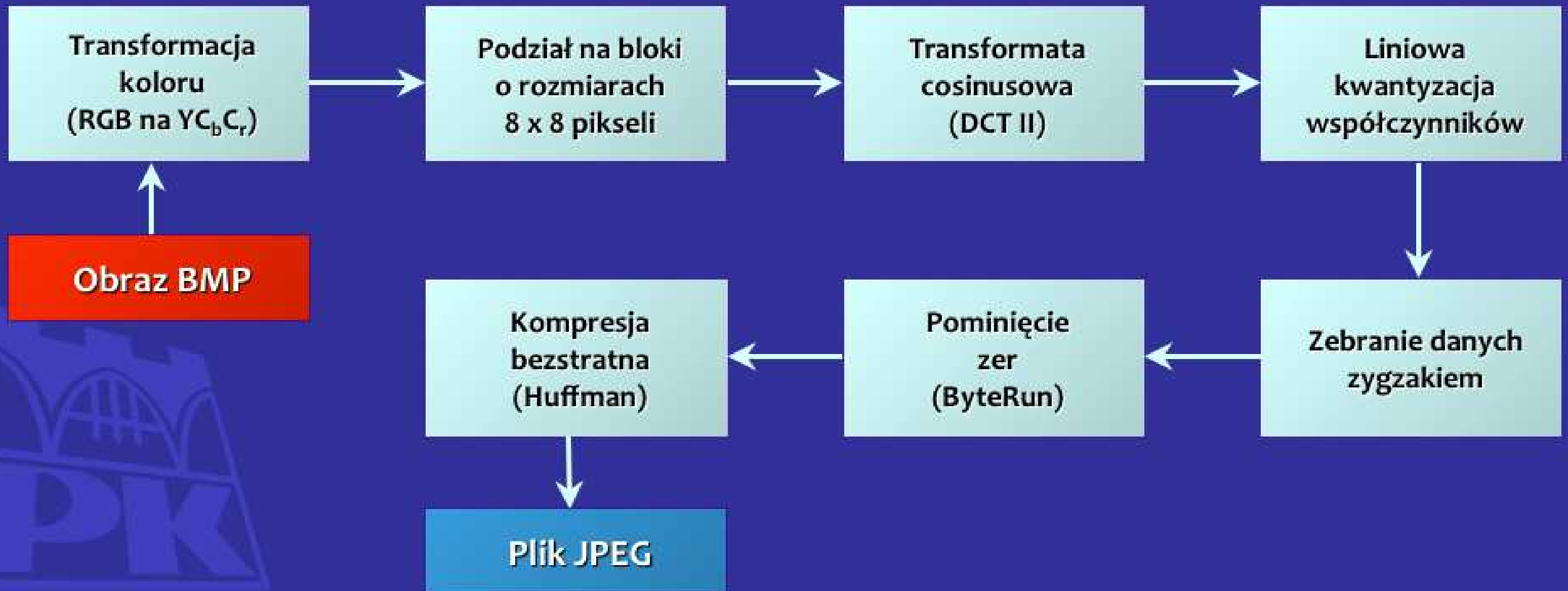
4. JPEG posiada dość rozbudowane środki informowania o absolutnej przestrzeni barw obrazu (możliwość umieszczenia profilu koloru w pliku).
5. JPEG obsługuje wyświetlanie progresywne (zwiększanie rozdzielczości obrazka w miarę napływania nowych danych).
6. JPEG jest przystosowany do transmisji w sieci – może być dekodowany przy sekwencyjnym dostępie do danych.

- Kroki algorytmu

- Transformacja przestrzeni barw (opcjonalnie)
- Podpróbkiwanie składowych chrominancji (opcjonalnie)
- Podział składowych na bloki 8x8 pikseli i zastosowanie transformaty DCT dla każdego bloku

- Kroki algorytmu

- Kwantyzacja macierzy współczynników transformaty na podstawie tabeli kwantyzacji
- Zebranie danych zygzakiem
- Kodowanie współczynników za pomocą kodów Huffmana lub arytmetycznie (rozszerzenie)
- Zapisanie zakodowanych danych w pliku o strukturze opisanej przez standard



- Transformacja przestrzeni barw (opcjonalnie)
 - Przestrzeń zawierające oddzielne składowe luminancji i chrominancji, np. $Y C_r C_b$
 - lub pominięcie transformacji i kompresja wprost obrazu multispektralnego (jak RGB, CMYK, do 255 składowych)
 - Kilka celów
 - dekorelacja składowych, które są kodowane niezależnie od siebie, zatem poprawa współczynników/jakości
 - umożliwienie kodowania różnych składowych z różną jakością

- Transformacja ta rozkłada obraz w kolorach RGB na składową luminancyjną (Y) i dwie składowe chrominancyjne (składowa antyczerwona C_r i antyniebieska C_b), podobnie jak to się dzieje w telewizji kolorowej.
- Składowe chrominancyjne mogą zostać poddane próbkowaniu ze zmniejszoną rozdzielczością.

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$

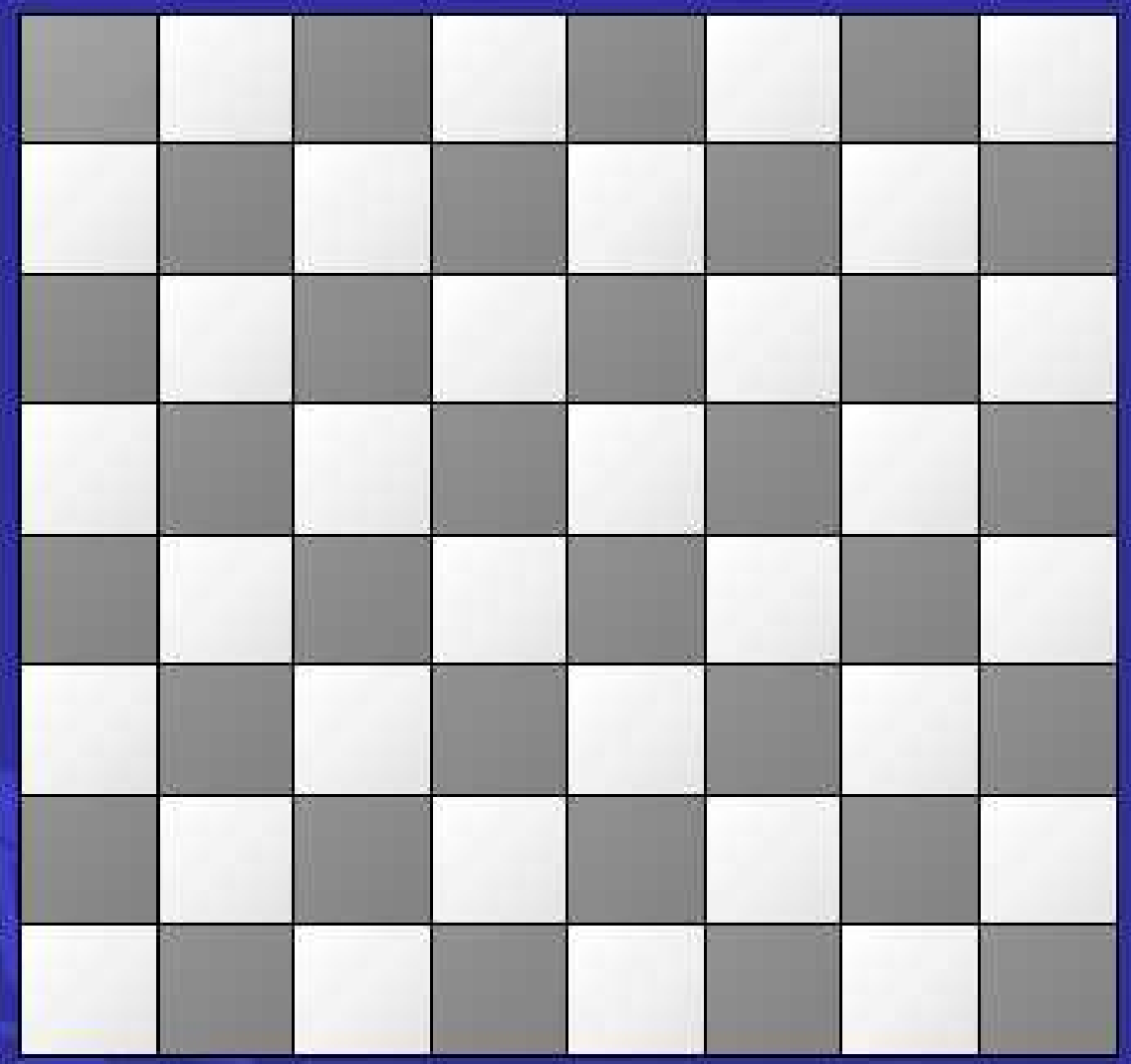
$$C_r = 128 - 0,168736 \cdot R - 0,331264 \cdot G + 0,5 \cdot B$$

$$C_b = 128 + 0,5 \cdot R - 0,418688 \cdot G - 0,081312 \cdot B$$

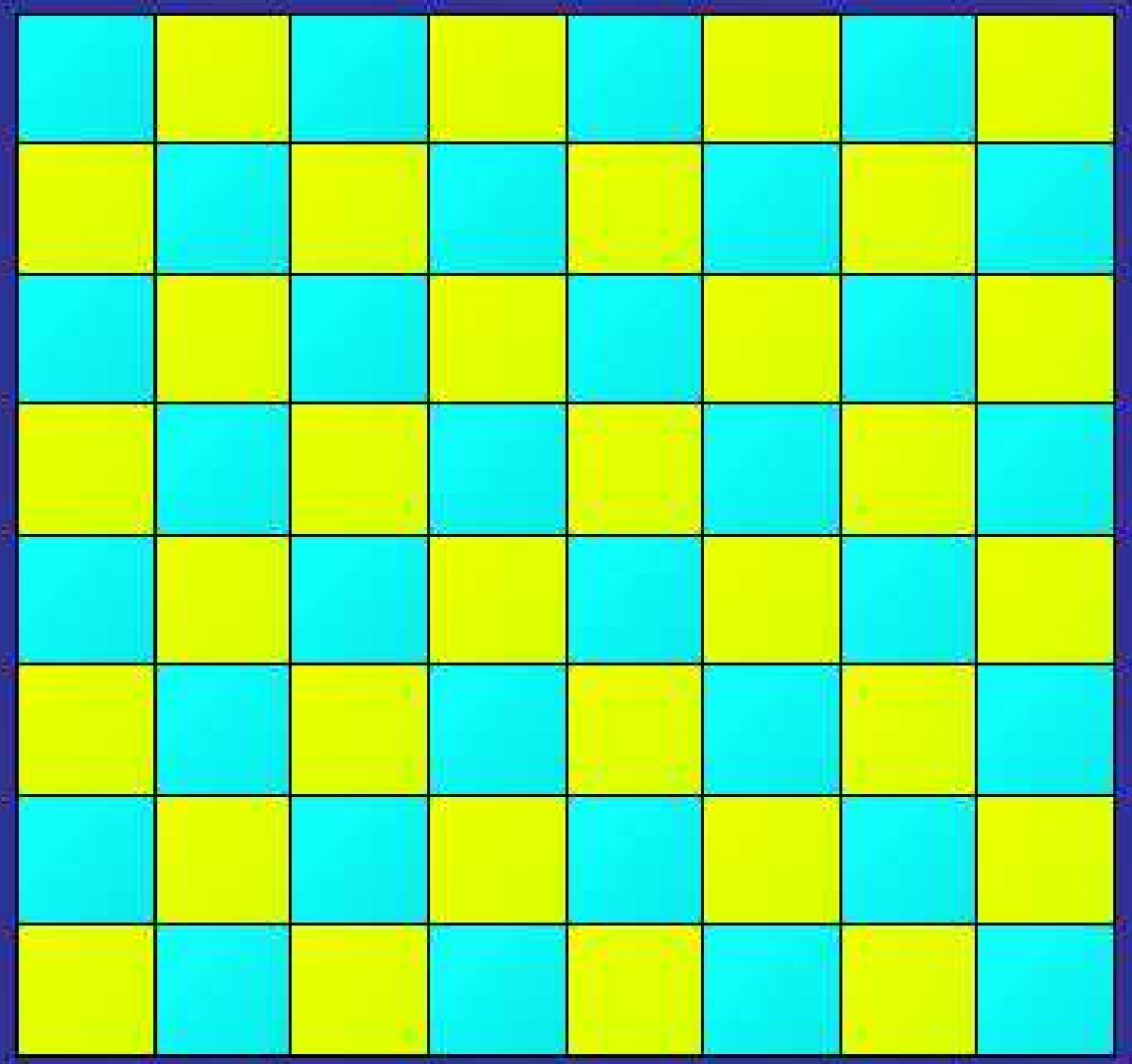


- **Podpróbkiwanie składowych chrominancji (opcjonalnie)**
 - Oko ludzkie jest znacznie bardziej czułe na jasność obrazu niż na kolor
 - zarówno pod względem rozdzielczości jak i głębi
 - Skoro kompresujemy stratnie to można „straty” wprowadzić z uwzględnieniem czułości oka
 - składowe chrominancji są podpróbkiwane, ich rozdzielczość jest 2-krotnie mniejsza od składowej luminancji (obraz komputerowy to dane już po wstępnym próbkiwaniu i kwantyzacji)

- W zależności od wymaganej jakości, składowe chrominancji mogą zostać:
 1. Pozostawione bez zmian, tzw. format 4:4:4.
 2. Rozdzielczość pozioma jest zmniejszana o połowę (format 4:2:2).
 3. Rozdzielczość pozioma i pionowa jest zmniejszana o połowę (format 4:2:0).

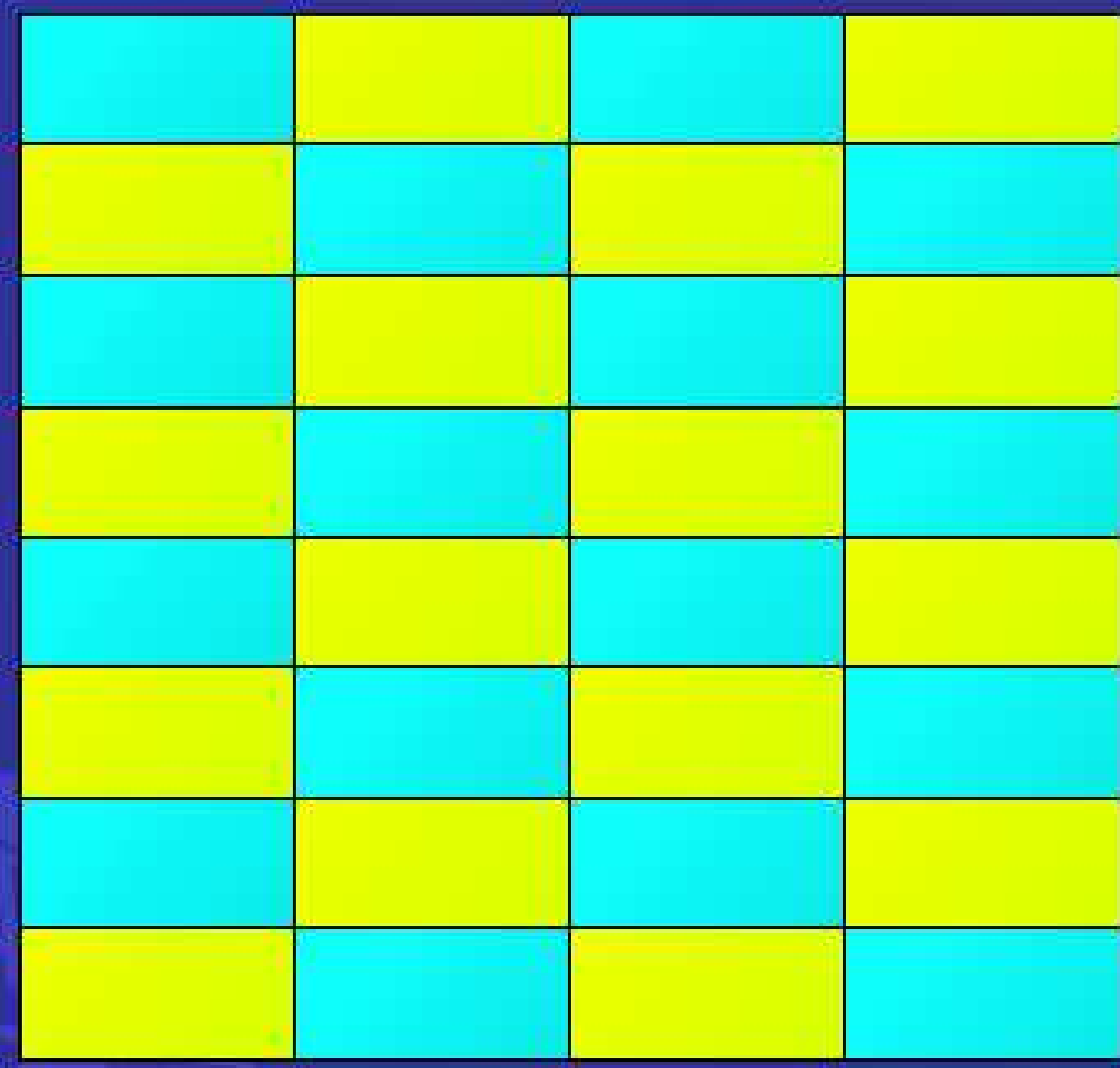


Y

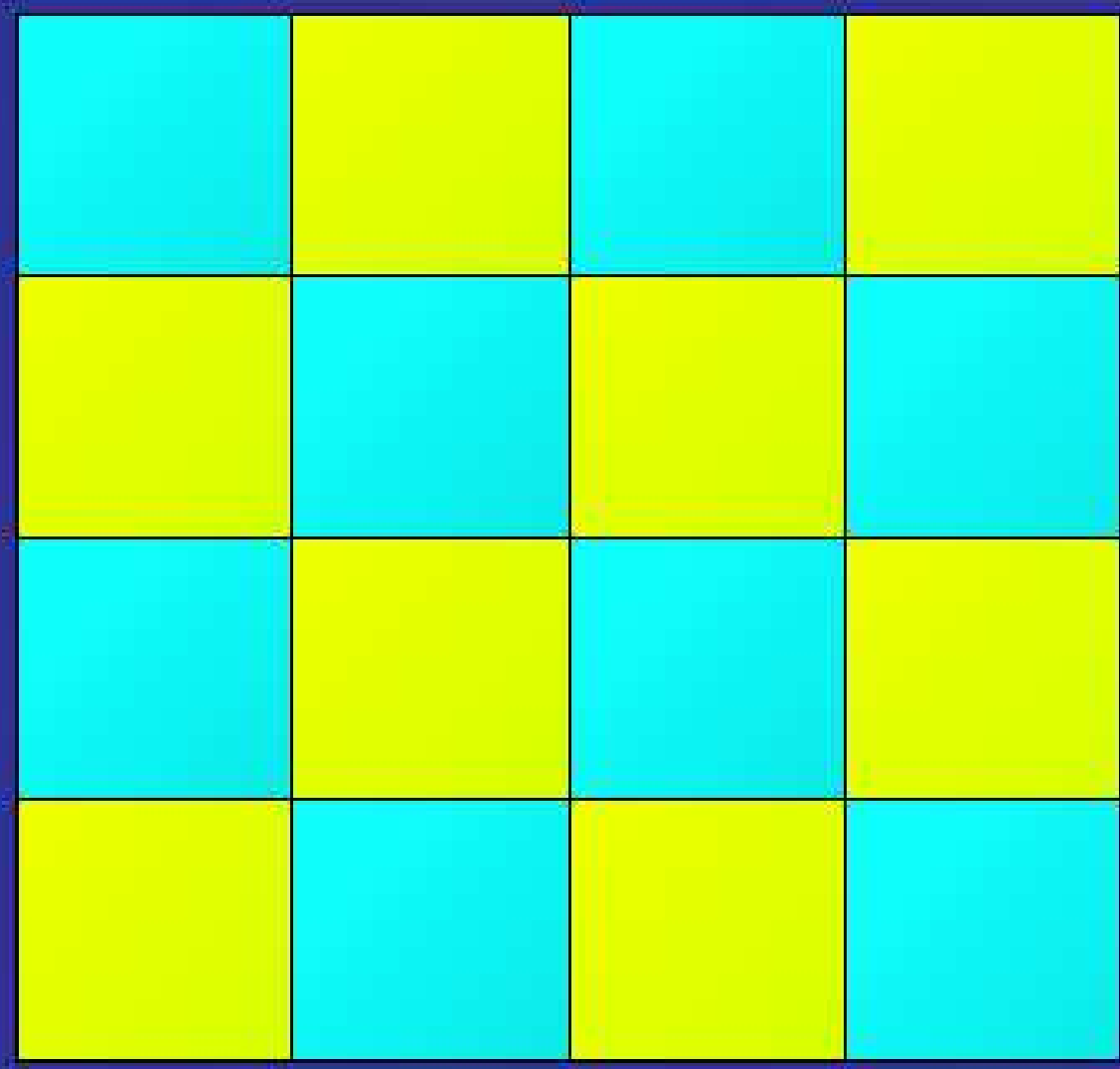


C_b i C_r 4:4:4

Transformacja przestrzeni barw



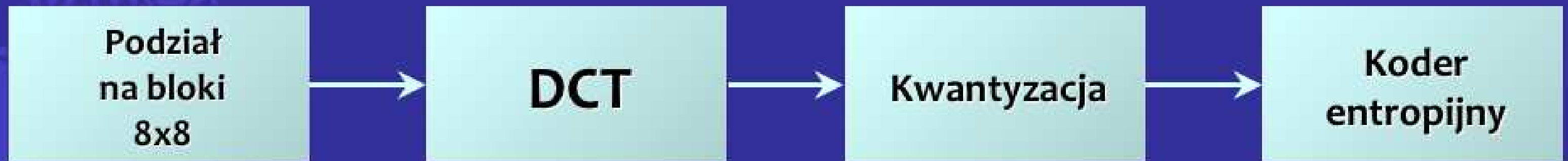
C_b i C_r 4:2:2



C_b i C_r 4:2:0

- **Kodowanie składowej**

- Podział składowych na bloki 8x8 pikseli
- Transformata DCT dla każdego bloku
- Kwantyzacja współczynników transformaty
- Kodowanie entropijne



- Każda ze składowych jest oddzielnie dzielona na bloki o rozmiarach 8 x 8 pikseli.
- Jeżeli składowe C_r i C_b były przeskalowane, bloki tych składowych obejmują obszar 16 x 8 lub 16 x 16 oryginalnego obrazka.
- Pewnym problemem jest uzupełnienie bloków leżących na prawej i dolnej krawędzi obrazu do całkowitej wielokrotności 8 (o ile rozmiary obrazu nie są taką wielokrotnością).

- Najgorszym rozwiązaniem jest uzupełnienie zerami, nieco lepszym – kolorem ostatniego piksela.
- Najlepsze jest dodanie takich pikseli, aby odwrotna transformata cosinusowa odtworzyła piksele należące do obrazka jak najbardziej zbliżone wartościami do oryginalnych.
- W praktyce wykonuje się to najczęściej dla trzech pierwszych współczynników transformaty.

- Generalnie rzecz biorąc **transformata**, to wyrażenie jakiejś funkcji jako (skończona lub nie) suma pewnego zestawu funkcji zwanego **bazą** transformaty.
- Np. baza transformaty Fouriera to zestaw sinusów i cosinusów o częstotliwości podstawowej i wielokrotnych.
- Współczynniki transformaty to po prostu zestaw liczb przez jakie mnożymy kolejne funkcje z bazy, aby uzyskać analizowaną funkcję pierwotną.
- W ogólnym przypadku współczynniki uzyskujemy licząc korelacje między kolejnymi funkcjami z bazy, a funkcją badaną w określonym przedziale.

```
static public String show...
String str = Integer.toString(12345);
int count = leadingZeros(count) - str.length();
```

```
process
variable X : int;
begin
X := 2 and 0;
after 10;
```

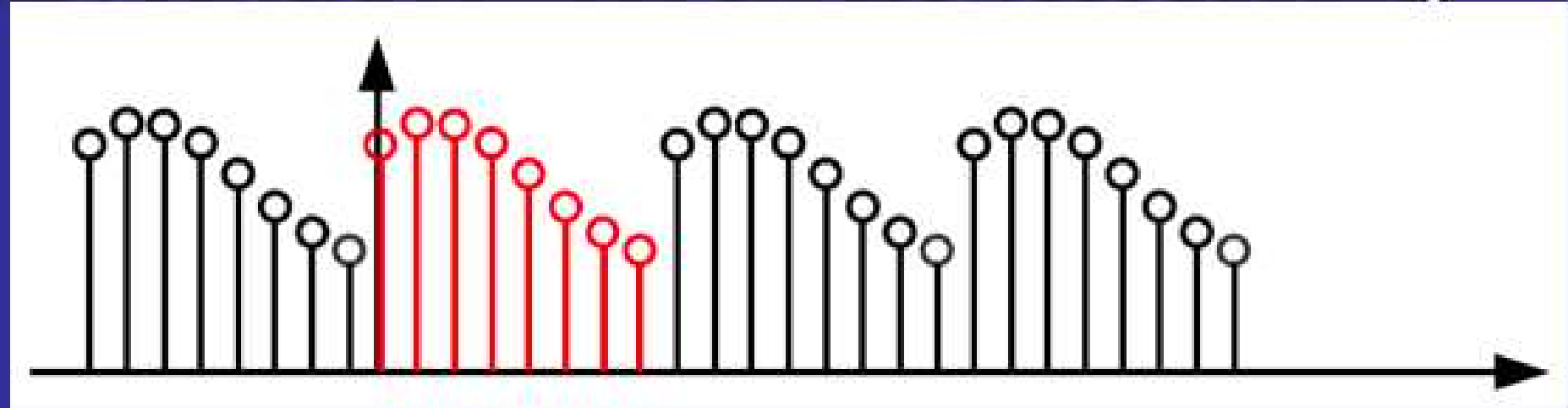
- W przypadku transformat dyskretnych, funkcja badana i funkcje bazy są zbiorami próbek, a korelacja jest iloczynem skalarnym funkcji bazowej i badanej.
- Żeby transformata miała sens, jej baza musi być **ortogonalna**.
- To znaczy, że korelacja dwóch dowolnych **różnych** funkcji z bazy jest równa zero.

```
static public String show...
String str = Integer.toString(12345);
int count = leadingZeros(count - str.length());
```

```
process
variable X : int;
begin
X := 2;
after TOR;
```

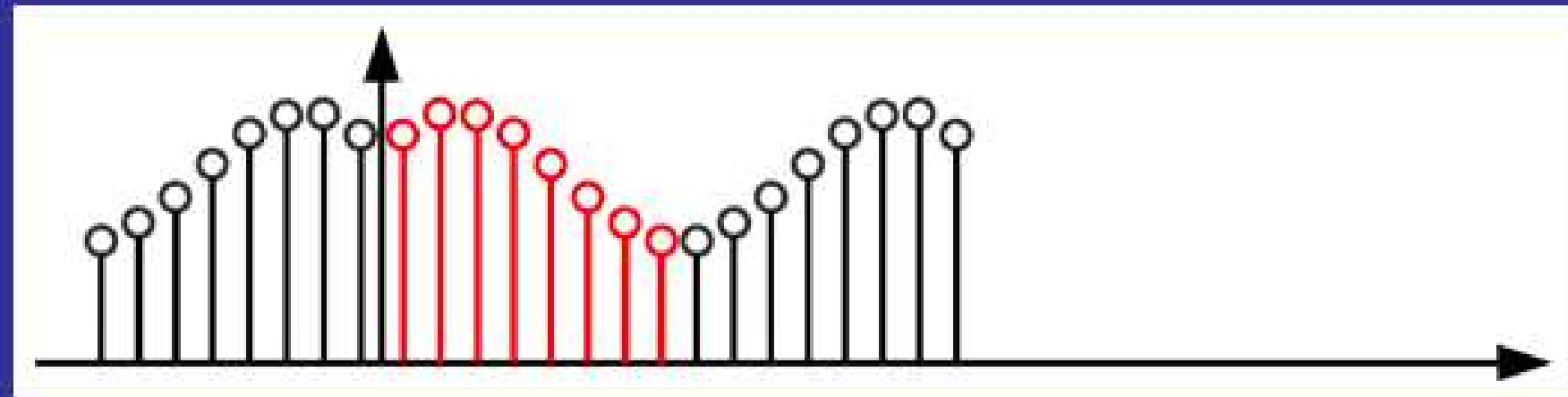
- Skąd to wymaganie? Jeżeli nie jest spełnione, to znaczy, że któraś z funkcji bazy może być wyrażona przez kombinację liniową innych funkcji bazy.
- Wskutek tego transformata nie jest jednoznaczna – badaną funkcję można poskładać z bazowych na kilka różnych sposobów.
- Nieortogonalność oznacza też, że albo niektóre z funkcji bazowych są zbędne, albo też transformata nie zadziała dla dowolnej funkcji badanej.

- Dyskretna transformata Fouriera zakłada okresowość badanej funkcji:



- Przez to funkcja z reguły jest nieciągła na granicach, co zwiększa wartości składowych o wyższych częstotliwościach. Oprócz tego każda składowa sinusoidalna ma inną fazę, dlatego transformata Fouriera jest zespolona.

- Transformata cosinusowa natomiast zakłada symetrię funkcji wokół osi Y (parzystość funkcji):



- Funkcja parzysta może być wyrażona przez sumę samych cosinusów, nieprzesuniętych w fazie. Dlatego transformata cosinusowa nie jest zespolona. Oprócz tego po lewej stronie badanego przedziału nie ma nieciągłości, więc amplitudy cosinusów o wyższych częstotliwościach są małe (tzw. cecha skupiania energii).

- Transformata cosinusowa ma w sumie 8 wariantów wynikających z następujących możliwości:
 - po prawej może nastąpić odbicie tylko po osi Y, albo po obu osiach.
 - symetria po lewej może być ustawiona na pierwszej próbce sygnału, albo w połowie między pierwszą próbką a jej lustrzanym odbiciem.
 - symetria po prawej też może być ustawiona na próbce lub między próbkami.

- JPEG używa transformaty typu II: odbicie po prawej tylko w osi Y, symetria po obu stronach między próbkami.
- Podobnie jak DFT posiada przyspieszony algorytm FFT, tak DCT posiada również (bazowany na FFT) algorytm przyspieszonego przeprowadzania transformaty.

- Transformata DCT dla bloku 8x8 pikseli

- DCT przeprowadzamy po sprowadzeniu nominalnego zakresu jasności pikseli do przedziału symetrycznego względem 0

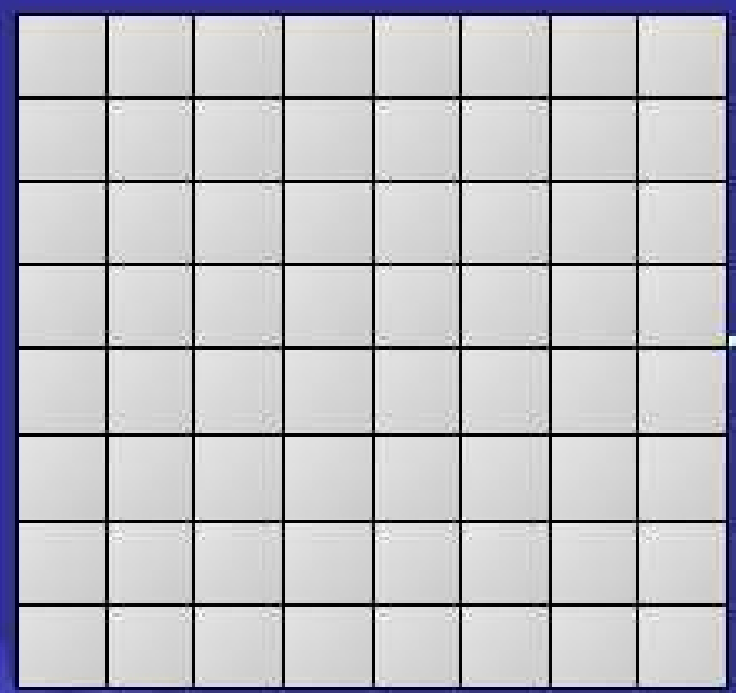
$$[0 \dots 2^N - 1] \rightarrow [-2^{N-1} - 1 \dots 2^{N-1} - 1]$$

- Współczynniki transformaty niskich częstotliwości zgrupowane są blisko lewego górnego rogu macierzy

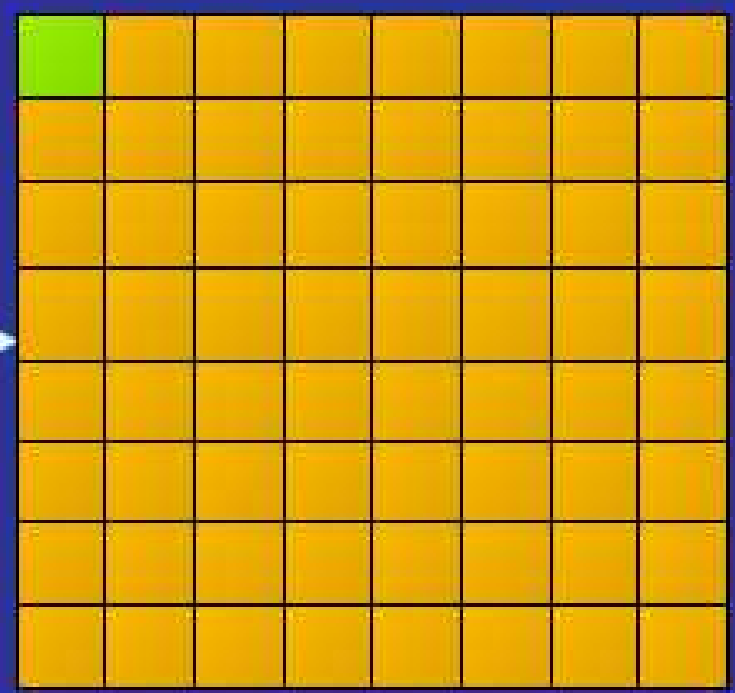
- Sam lewy górny róg to składowa stała jasności bloku

```
static public String show...  
String str = Integer.toString(ASCII);  
int iCount = leadingZeros(count) - str.length();  
int iCount = leadingZeros(count) - 1;
```

```
begin  
variable X : int;  
X := 7; read B;  
after 100  
end
```



DCT



-  - DC „direct current”
-  - AC „alternate current”

- Kwantyzacja macierzy współczynników transformaty na podstawie tabeli kwantyzacji (o rozmiarze 8x8)
 - po DCT, dla składowej o głębi N bpp, współczynniki transformaty wymagają $N+3$ bitów
 - czyli najpierw mamy ekspansję danych
 - typowo większość współczynników AC ma małe wartości ...
 - ... tym mniejsze im wyższej częstotliwości odpowiadają ...
 - ... i małe znaczenie dla percepcji obrazu
 - każdy z współczynników transformaty DCT dzielimy przez odpowiadający mu element tabeli kwantyzacji ...

- Kwantyzacja macierzy współczynników transformaty na podstawie tabeli kwantyzacji (o rozmiarze 8x8) — c.d.
 - tabela kwantyzacji jest parametrem algorytmu
 - komitet JPEG przygotował tabele standardowe dobrane na podstawie eksperymentów z rzeczywistymi obrazami oraz rzeczywistymi urządzeniami akwizycji
 - inne dla luminancji (np. →)
inne dla chrominancji
 - dzieląc wszystkie współczynniki przez stałą wartość kontrolujemy współczynnik kompresji

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

- Kodowanie współczynników za pomocą kodów prefiksowych
 - Kodujemy odmiennie współczynniki DC i AC, oba rodzaje współczynników z użyciem kodów prefiksowych
 - wygenerowanych algorytmem Huffmana (w dość złożony sposób — szczegóły: patrz artykuł Wallace’a)
 - Współczynniki DC
 - kodujemy różnice współczynników DC kolejnych bloków
 - Współczynniki AC
 - ciąg współczynników AC bloku zwykle zawiera długie podciągi zer; sufiks ciągu zwykle jest długim ciągiem zer
 - kodujemy wariantem algorytmu RLE połączonym z kodowaniem Huffmana

- Dlaczego JPEG jest stratny?
 - przede wszystkim
 - podpróbkowanie składowych chrominancji
 - kwantyzacja
 - również „nieodwracalne”
 - transformacja przestrzeni barw
 - DCT

- **Rozszerzenia standardu**
 - tryb progresywny względem jakości
 - tryb hierarchiczny (piramidowy)
 - kodowanie arytmetyczne
(QM-Coder zamiast algorytmu Huffmana)
 - inne (praktycznie nie stosowane)

- **Nowy standard komitetu JPEG dla stratnej i bezstratnej kompresji obrazów**
 - **ITU-T; ISO/IEC: Information technology – JPEG 2000 image coding system: Core coding system. ITU-T Recommendation T.800 and ISO/IEC International Standard 15444-1, August 2002.**



- Z roku 2000, następca algorytmu JPEG i Lossless JPEG, wyłoniony w drodze konkursu (ogłoszonego w 1997) oparty o algorytm CREW (Zandi, Allen, Schwartz, Boliek)
 - Algorytm CREW był zgłoszony w odpowiedzi na konkurs, który miał wyłonić algorytm JPEG-LS. Jako podstawę JPEG-LS wybrano LOCO-I, lecz bogactwo dodatkowych własności algorytmu CREW (innych niż stratna/bezstratna kompresja obrazu jako całości) skłoniło komitet do rozpoczęcia prac nad JPEG2000.
 - Christopoulos, C.; Skodras, A.; Ebrahimi, T.: The JPEG2000 Still Image Coding System an Overview. IEEE Transactions on Consumer Electronics, November 2000, Vol. 46(4), pp. 1103-27.

- **Cechy algorytmu**

- Algorytm stratnej/bezstratnej kompresji obrazów multispektralnych, barwnych oraz w stopniach szarości oparty o transformatę falkową
- Dla kompresji stratnej i bezstratnej, dostępne możliwości
 - kodowanie progresywne, w tym progresja: stratne → bezstratne
 - kodowanie piramidowe
 - kodowanie progresywne względem ROI (region of interest)
 - dostęp swobodny do fragmentów obrazu

- Cechy algorytmu

- Algorytm stratnej/bezstratnej kompresji obrazów multispektralnych, barwnych oraz w stopniach szarości oparty o transformatę falkową

- Dla stratnej

- lepsza jakość od JPEG przy tym samym współczynniku dla współczynników poniżej 0.25 bpp znacznie lepsza
- ROI w standardzie (w JPEG formalnie również było jako extension)
- niezła odporność na błędy transmisji
- poprawienie współczynnika (kosztem pogorszenia jakości) nie wymaga pełnego dekodowania

- Kodowanie predykcyjne
- Kwantyzacja wektorowa
- Kompresja fraktalna



Dziękuję za uwagę...

```
static public String show...
String str = Integer.toString(1456);
int count = leadingZeros(count - str.length());
    count = (count + 1);
```

```
process
variable X : int;
begin
X := 2 and 0;
    after 10;
end
```



Damian Grela
e-mail: dgrela@pk.edu.pl
<http://www.dgrela.pl>

