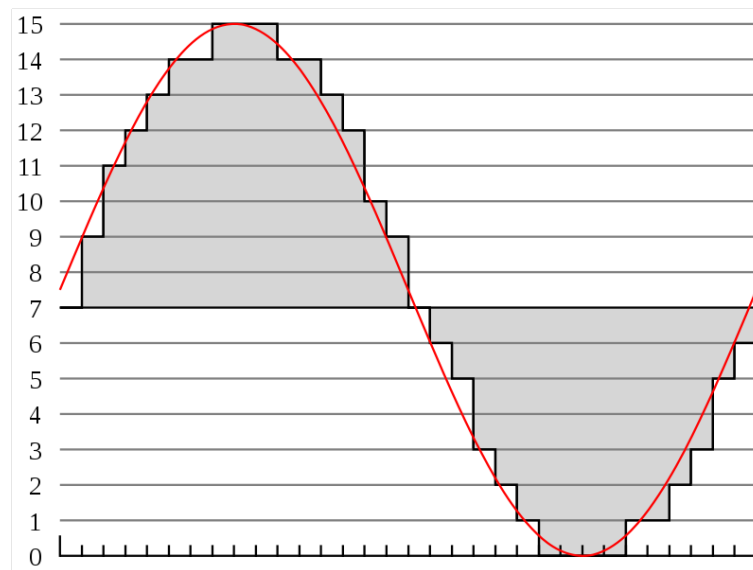
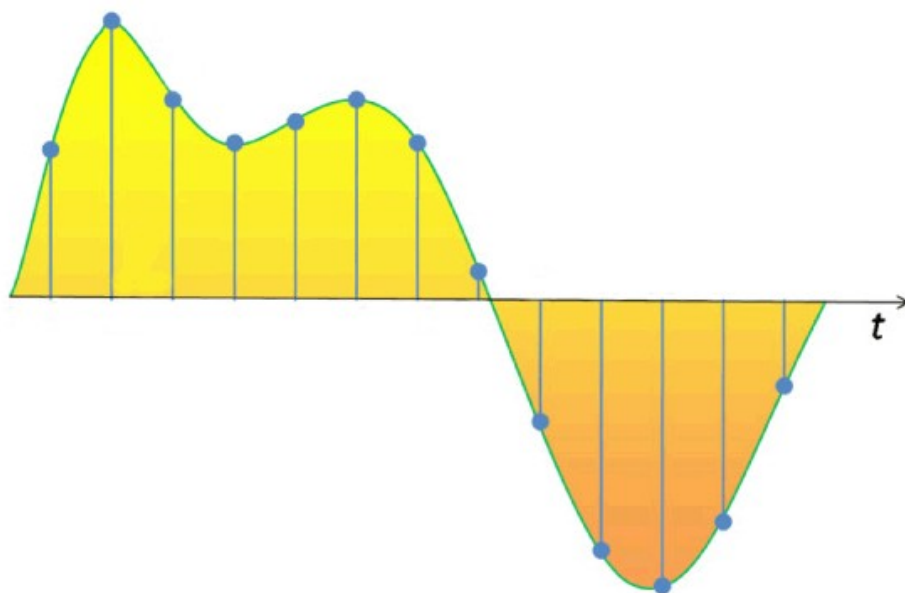


Цвет

URL: <http://www.school30.spb.ru/cgsg/cgc/>

E-mail: CGSG@yandex.ru

- Дискретизация сигнала – разбиение непрерывного сигнала на «выборки» (**sampling**, *sampling rate*)
- Квантование выборки – кодирование аналогового сигнала в дискретные величины (**quantization**)





8x8



16x16



32x32



64x64



128x128



256x256



2 цвета



3 цвета



4 цвета



8 цветов

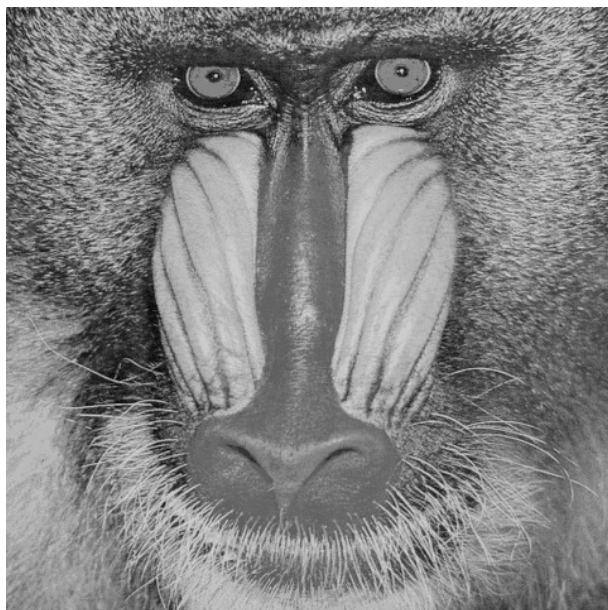


16 цветов

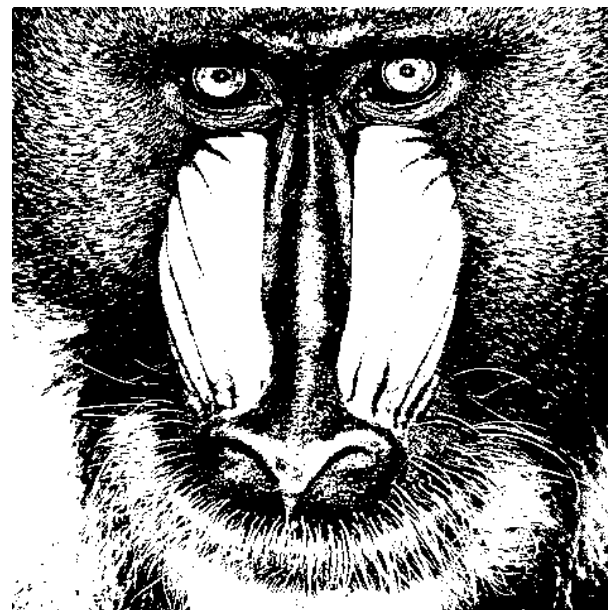


256 цветов

```
if (Img(x, y) > Threshold)
    color = 1;
else
    color = 0;
```

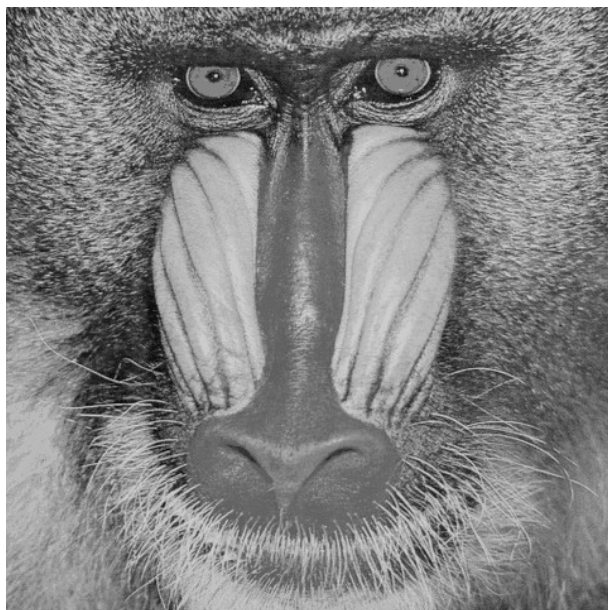


оригинал

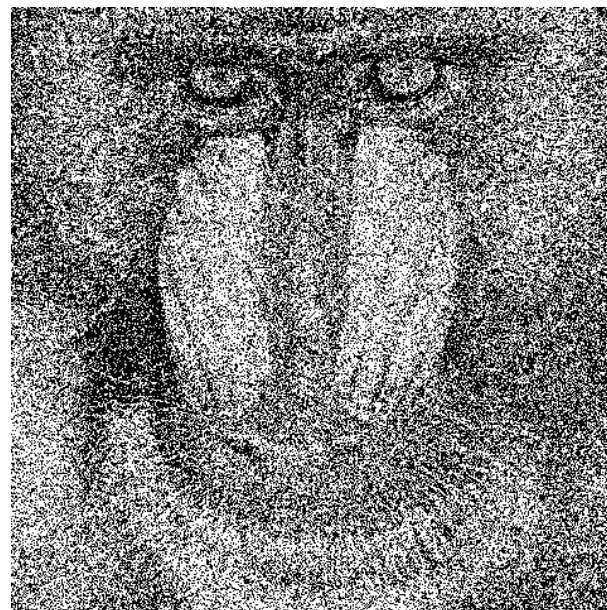


порог = 128

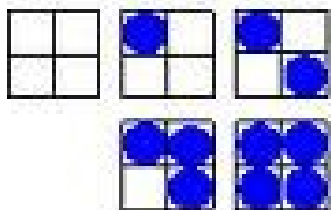

```
if (Img(x, y) > rand() % 255)
    color = 1;
else
    color = 0;
```



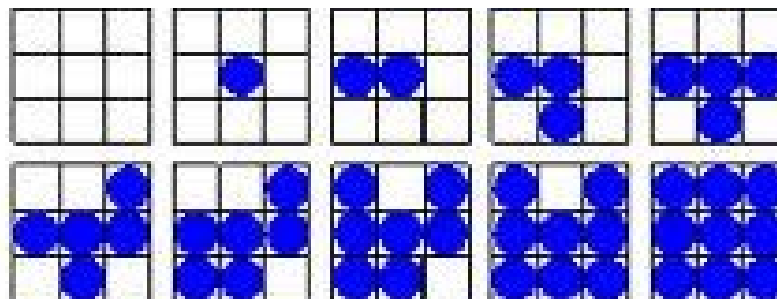
оригинал



«случайный» порог



5 уровней
(2x2)



10 уровней
(3x3)

0 2
3 1

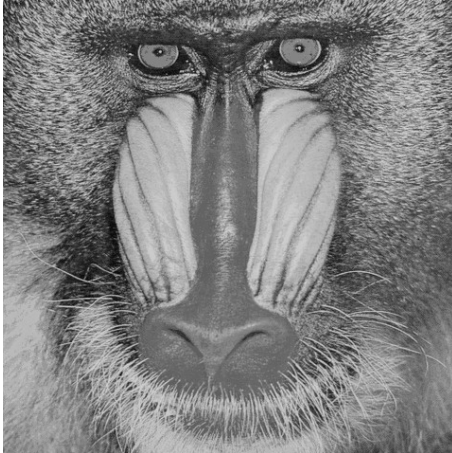


0	2	0	2		
3	1	3	1		
0	2	0	2		
3	1	3	1		

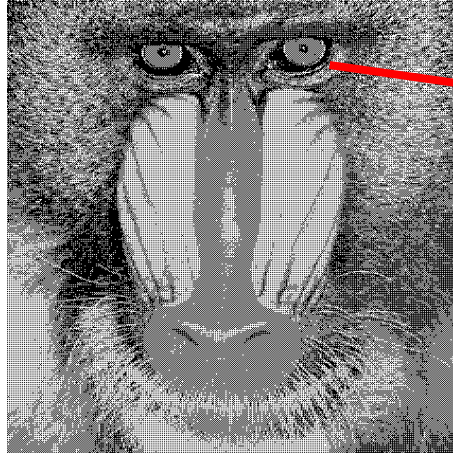
экран
заполняется
матрицами

```
int M_2x2[2][2] =
{
    {0, 2},
    {3, 1}
};

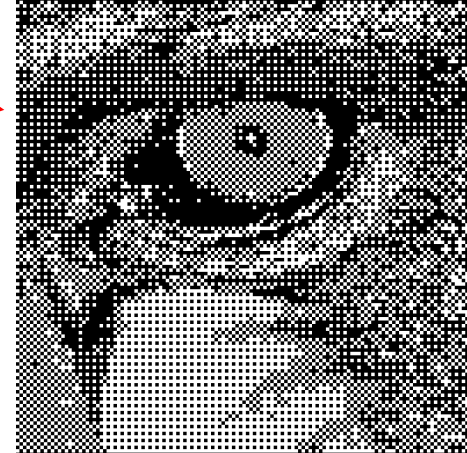
if (Img(x, y) * 5 / 256 > M_2x2[y % 2][x % 2])
    color = 1;
else
    color = 0;
```



оригинал



матрица 2x2

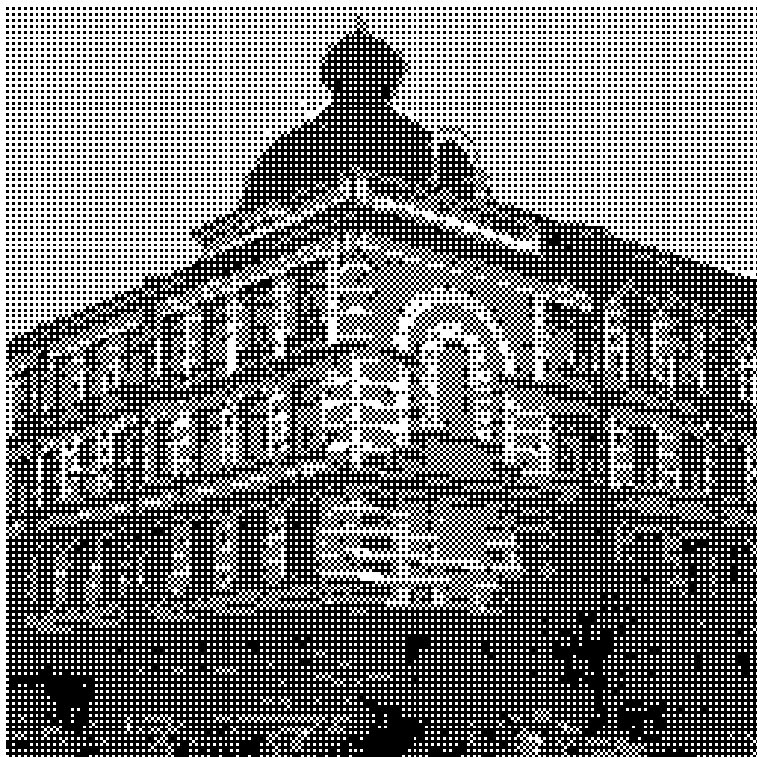


увеличенный
фрагмент

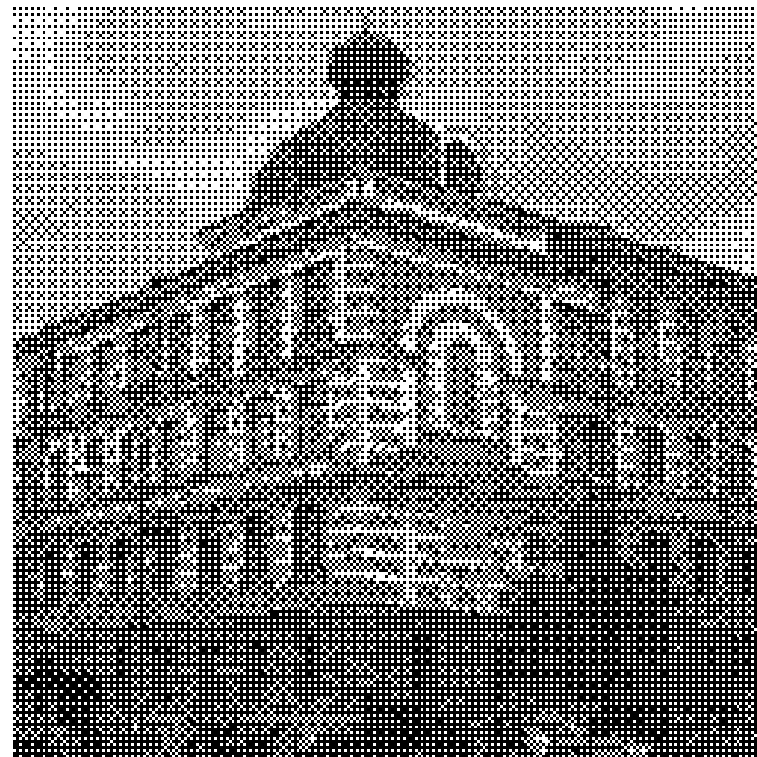

```
int M2[2][2] =
{
    {0, 2},
    {3, 1}
};

int M4[4][4] =
{
    { 0, 8, 2, 10},
    {12, 4, 14, 6},
    { 3, 11, 1, 9},
    {15, 7, 13, 5},
};

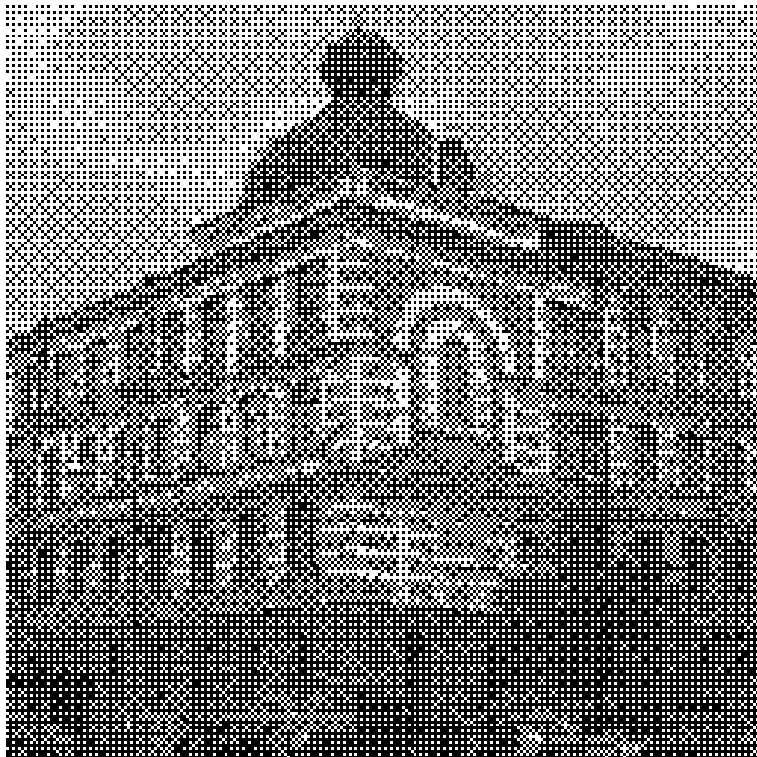
-----
for (i = 0; i < 4; i++)
    for (j = 0; j < 4; j++)
    {
        M2n[0 + i][0 + j] = Mn[i][j] * 4 + 0;
        M2n[0 + i][4 + j] = Mn[i][j] * 4 + 2;
        M2n[4 + i][0 + j] = Mn[i][j] * 4 + 3;
        M2n[4 + i][4 + j] = Mn[i][j] * 4 + 1;
    }
```



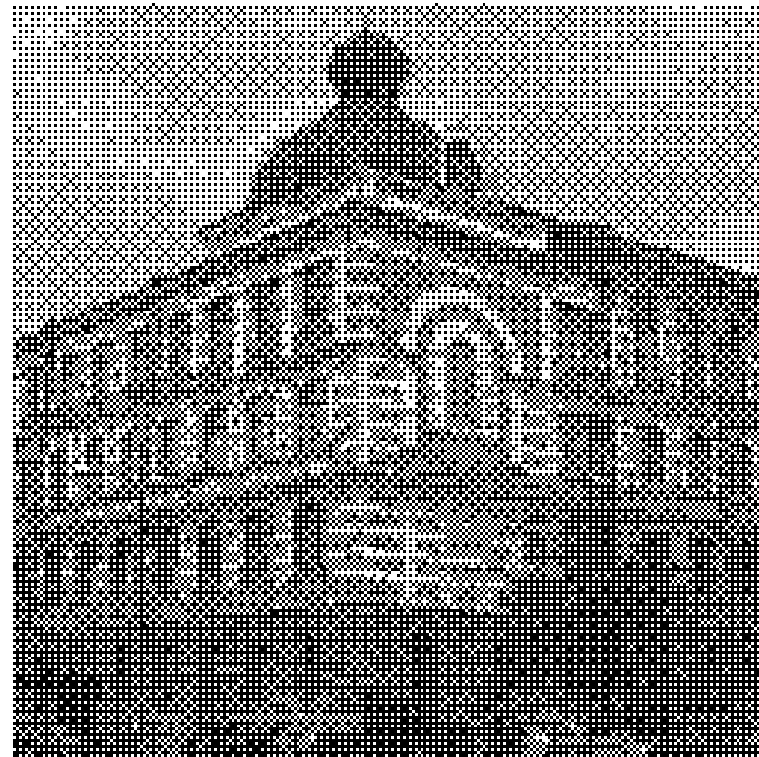
2x2



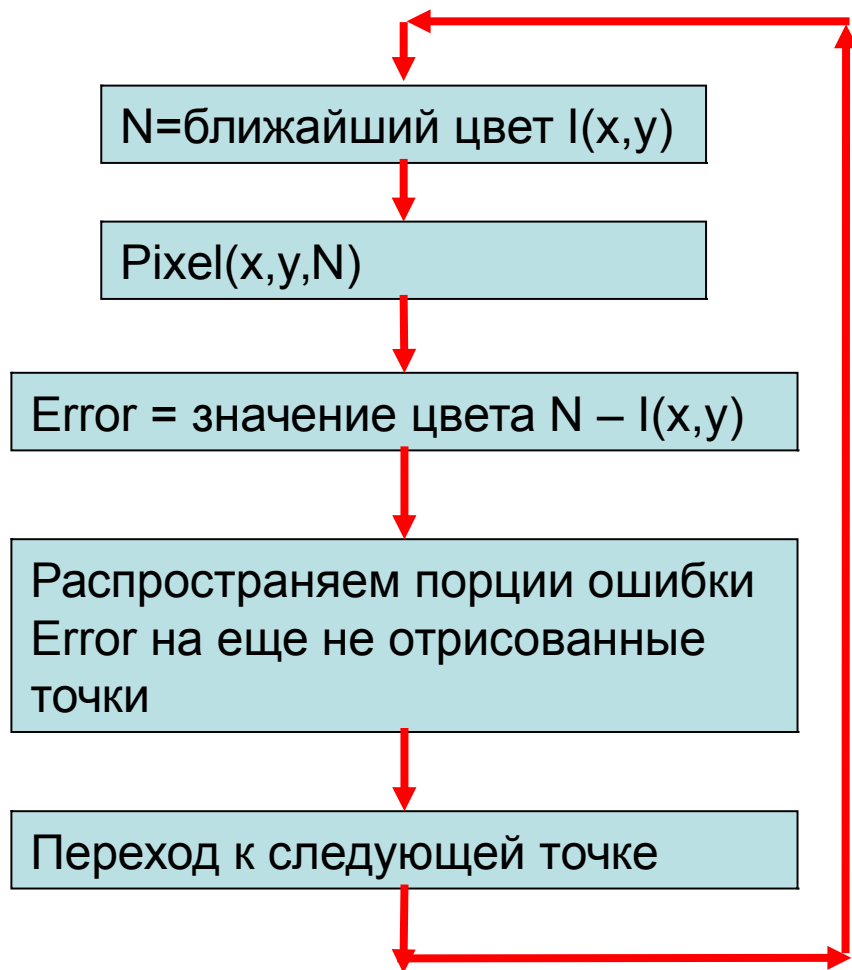
4x4



8x8

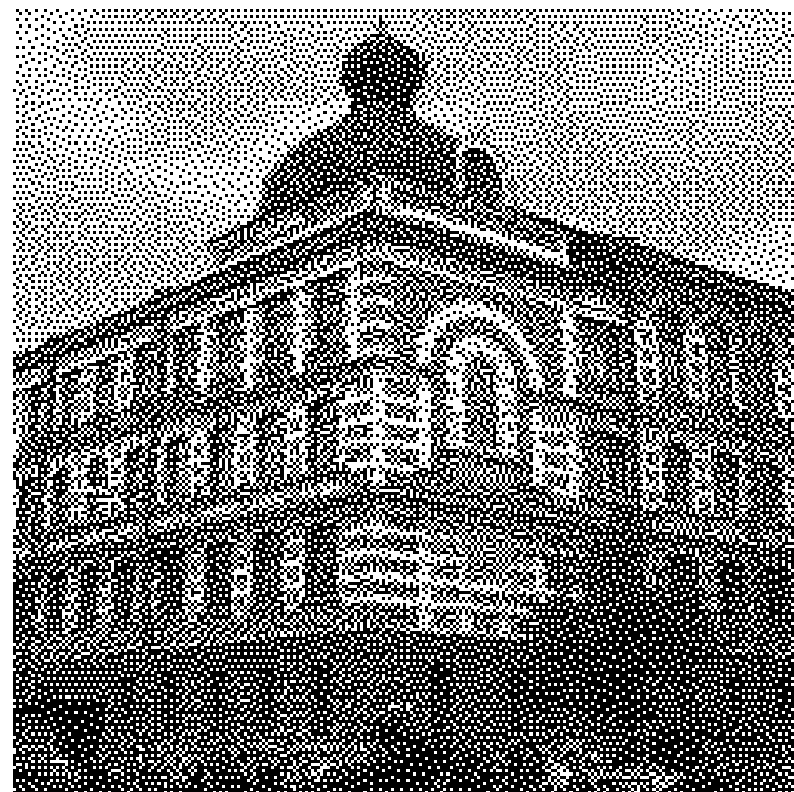


16x16



	*	7
3	5	1

1/16



*	3
3	2

False Floyd-Steinberg

1/8

		*	7	5
3	5	7	5	3
1	3	5	3	1

Jarvice, Judice, Ninke

1/48

		*	8	4
2	4	8	4	2
1	2	4	2	1

Stucki

1/42

		*	8	4
2	4	8	4	2

Burkes

1/32

Frankie Sierra

		*	5	3
2	4	5	4	2
	2	3	2	

1/32

		*	4	3
1	2	3	2	1

1/16

	*	2
1	1	

1/4

Универсальная палитра для любых изображений:

цвет задается по RGB каналам:

$$\text{ColorNo} = B + \text{SizeB} * (G + \text{SizeG} * R)$$



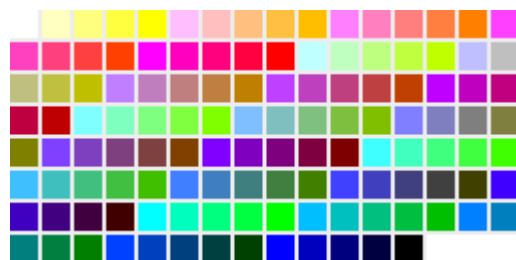
8



27

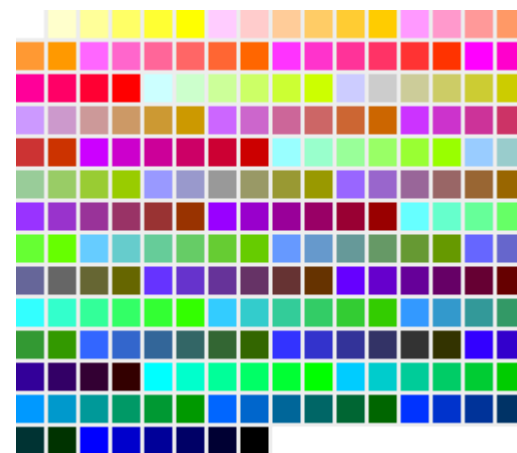


64

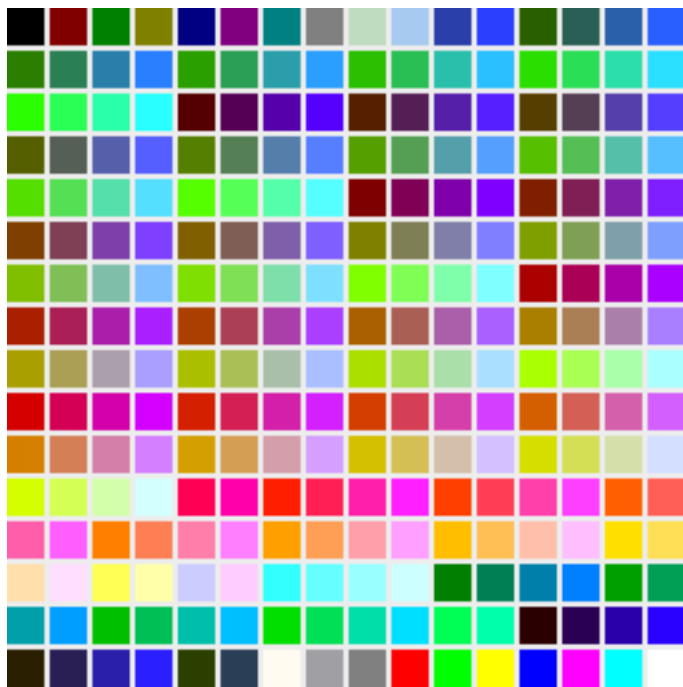


125

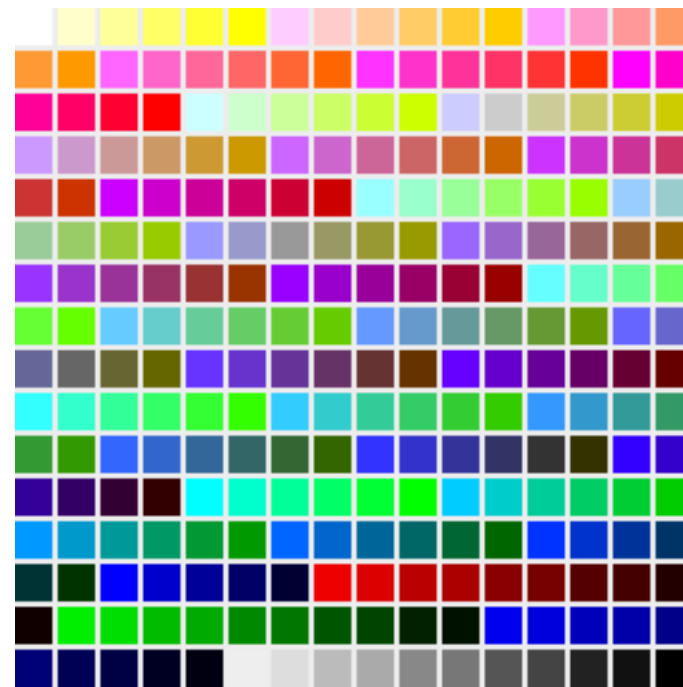
216



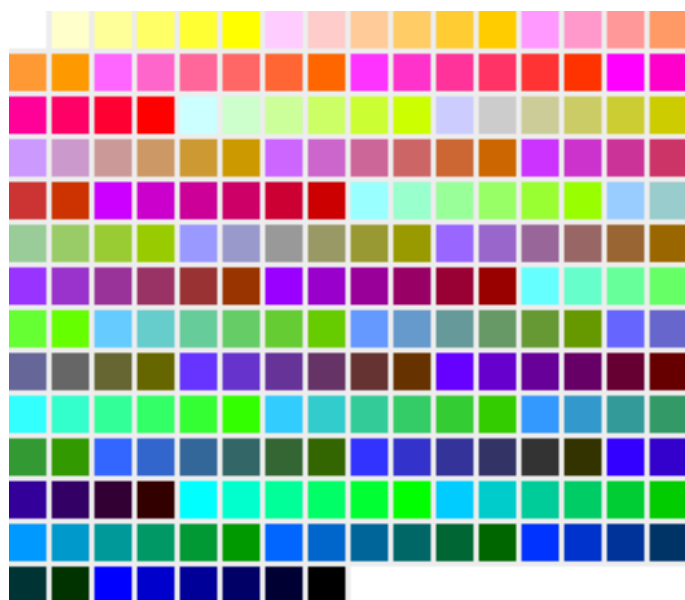
Используются в индексированных графических режимах



MS Windows



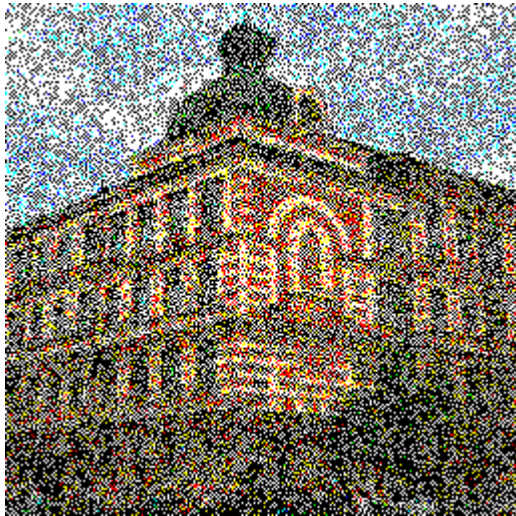
Mac OS



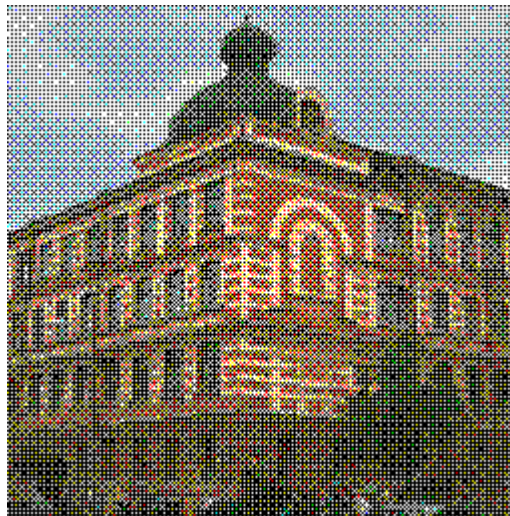
WEB палитра



оттенки по каналам
шаг: 0-51-102-163-204-255



random threshold

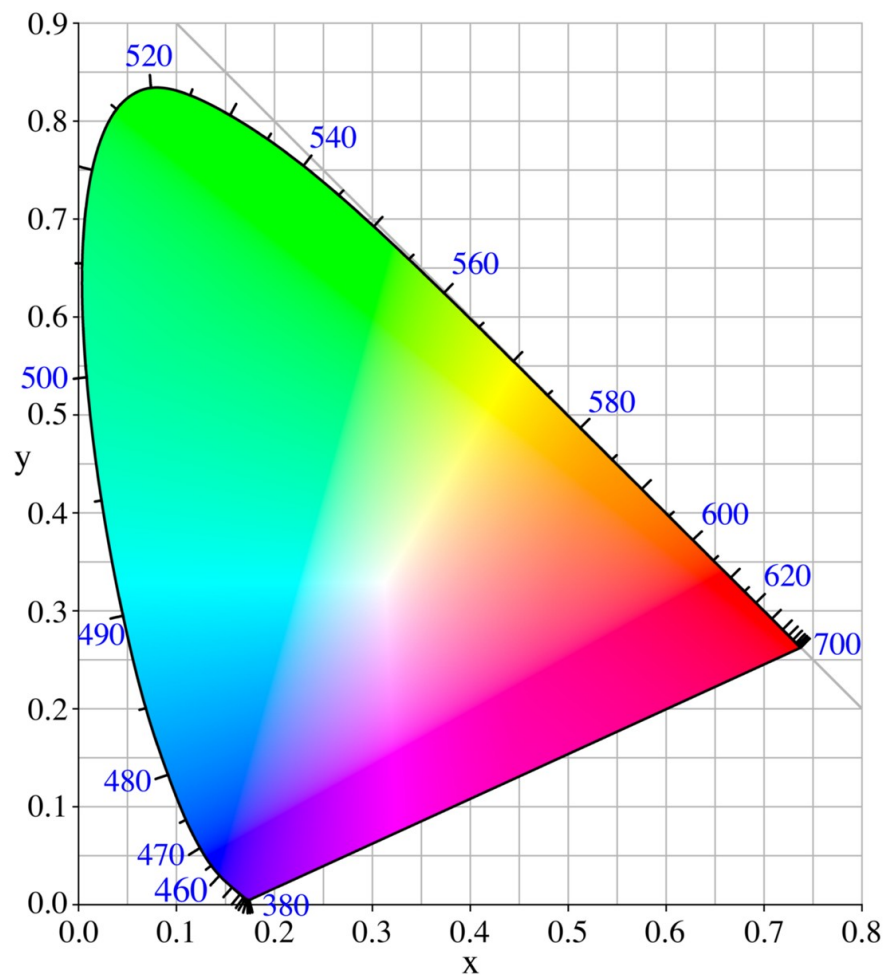


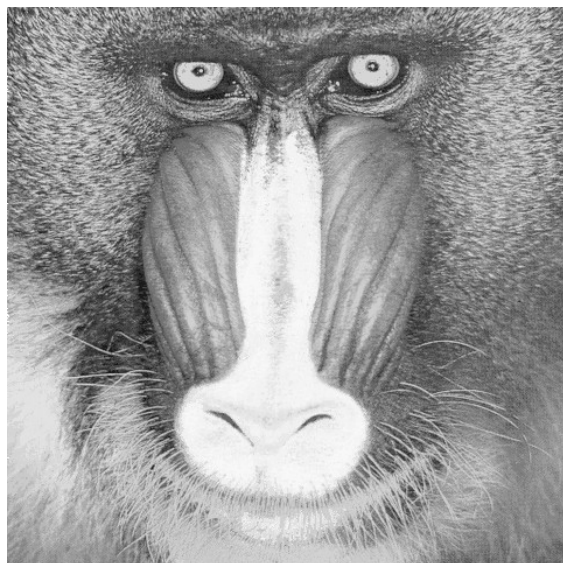
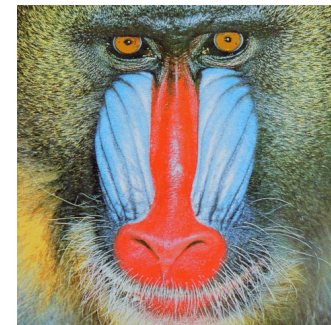
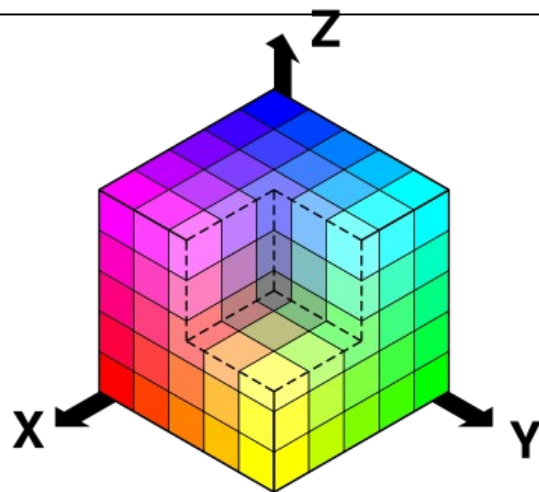
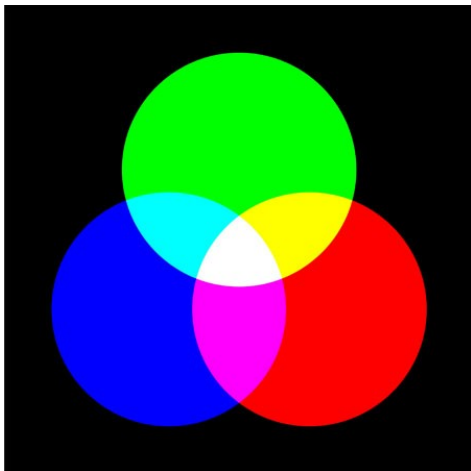
ordered dither



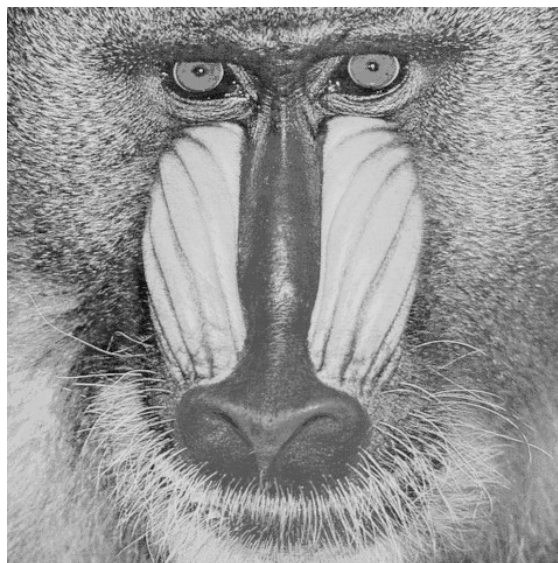
error diffusion

Международная Комиссия по Освещенности (Commission internationale de l'éclairage - CIE)

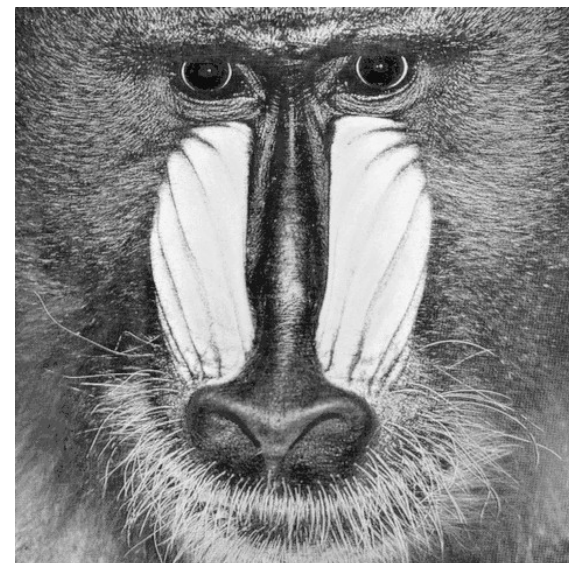




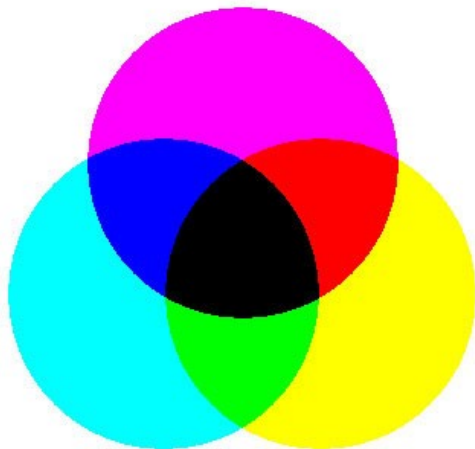
red



green



blue

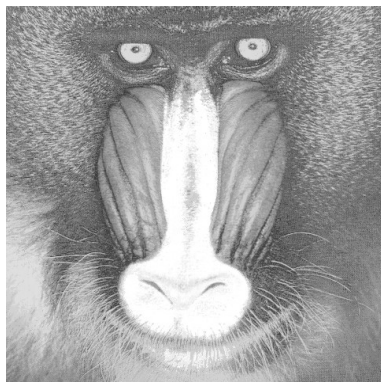
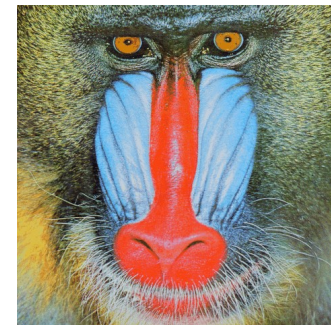


RGB2CMYK:

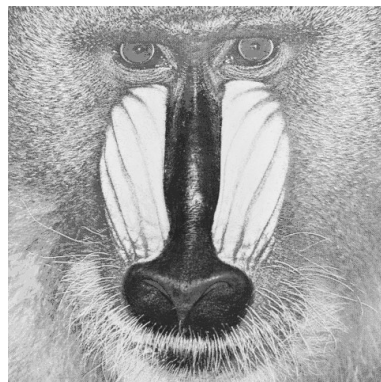
```
C = 1.0 - R;
M = 1.0 - G;
Y = 1.0 - B;
K = min(C, M, Y);
C = C - K;
M = M - K;
Y = Y - K;
```

CMYK2RGB:

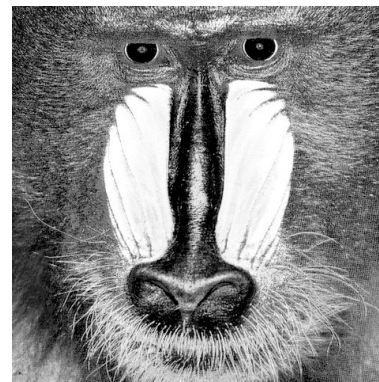
```
R = 1.0 - min(1.0, C + K);
G = 1.0 - min(1.0, M + K);
B = 1.0 - min(1.0, Y + K);
```



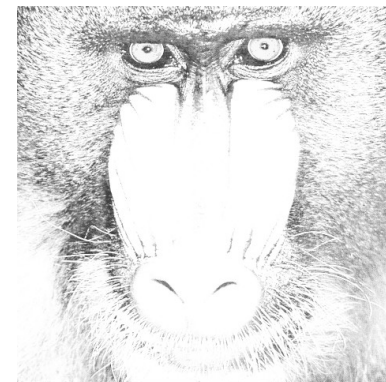
cyan



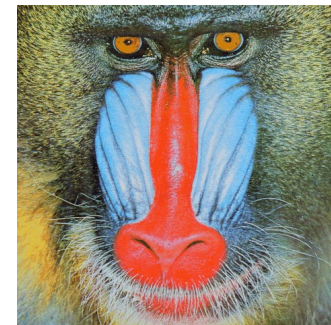
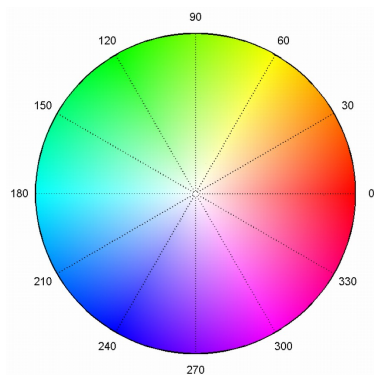
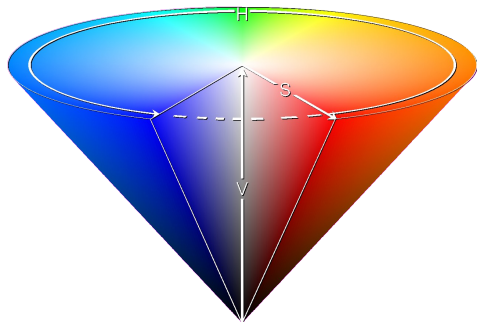
magenta



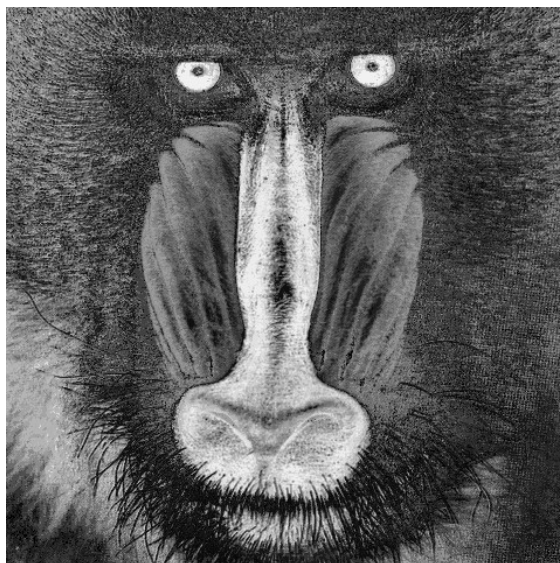
yellow



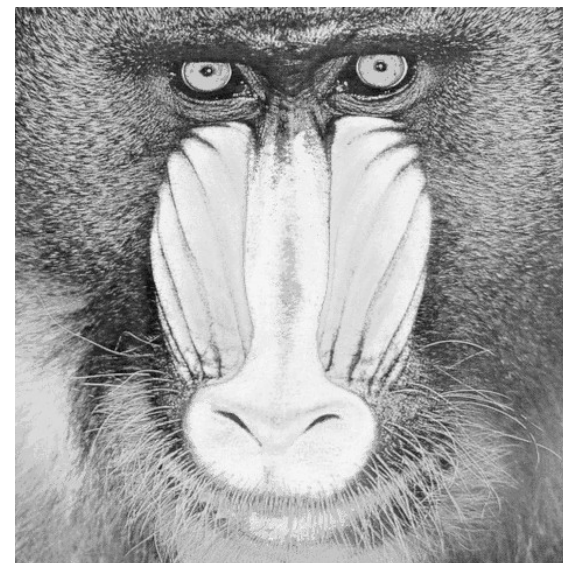
black



hue



saturation



value

HSV2RGB:

```

if (S == 0)
    return <V, V, V>;
else
{
    H = H / 60.0;
    n = (int)H;
    frac = H - n;
    c1 = V * (1.0 - S);
    c2 = V * (1.0 - S * frac);
    c3 = V * (1.0 - S * (1.0 - frac));
    if (n == 0)
        return <V, c3, c1>;
    if (n == 1)
        return <c2, V, c1>;
    if (n == 2)
        return <c1, V, c3>;
    if (n == 3)
        return <c1, c2, V>;
    if (n == 4)
        return <c3, c1, V>;
    if (n == 5)
        return <V, c1, c2>;
}

```

RGB2HSV

```

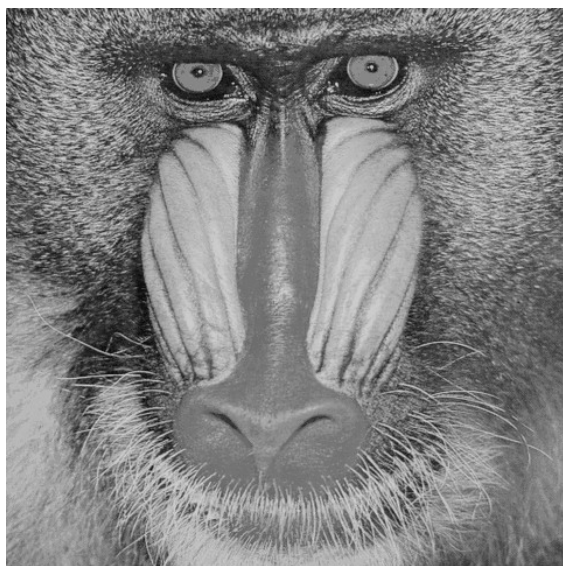
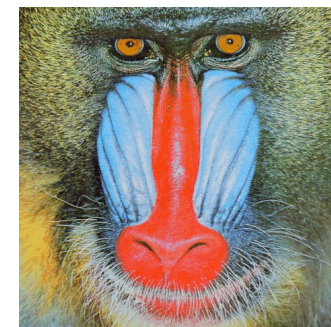
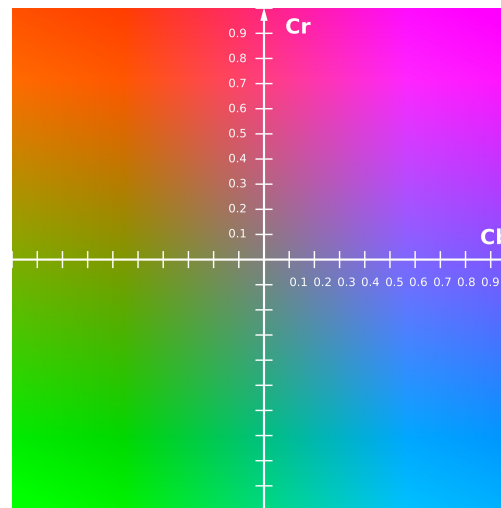
maxc = max(R, G, B);
minc = min(R, G, B);
delta = maxc - minc;

S = 0;
if (maxc > 0)
    S = delta / maxc;
V = maxc;
if (S == 0)
    H = 0; /* неопределено */
else
{
    rc = (maxc - R) / delta;
    gc = (maxc - G) / delta;
    bc = (maxc - B) / delta;
    if (R == maxc)
        H = bc - gc; /* Y-M */
    else if (G == maxc)
        H = 2 + rc - bc; /* C-Y */
    else
        H = 4 + gc - rc; /* M-C */
    H = H * 60.0;
}

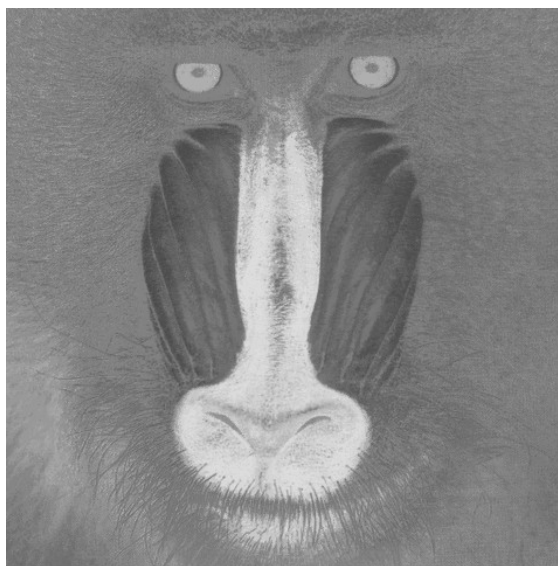
```


$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

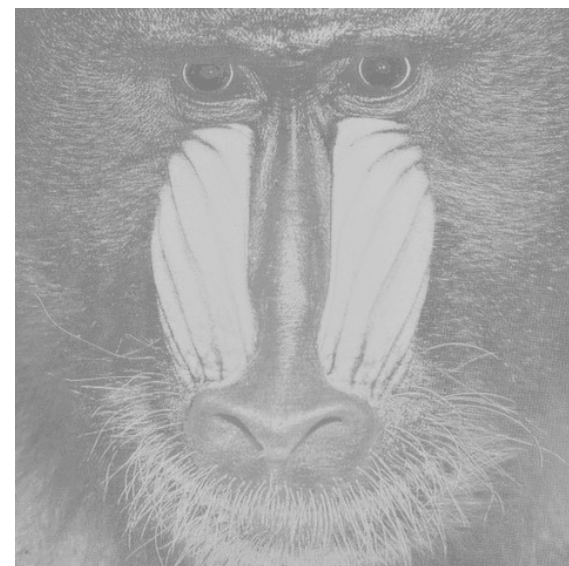
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} * \begin{bmatrix} Y \\ Cb - 128 \\ Cr - 128 \end{bmatrix}$$



Y



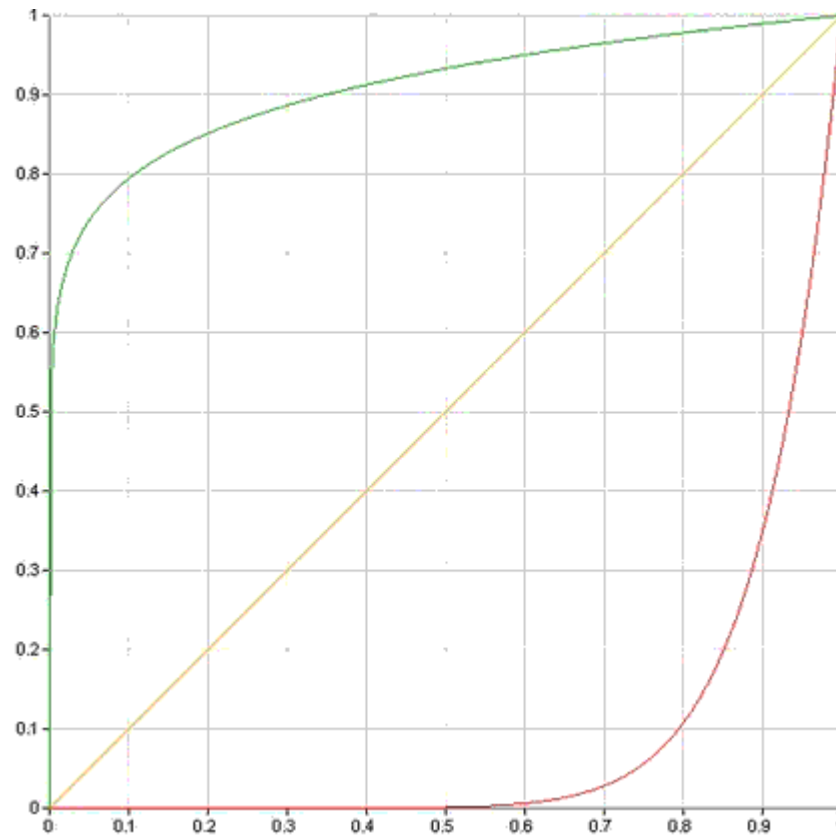
Cr



Cb

- LUT: `Color = LUT[Color]` ;
- Гамма коррекция, контрастность, яркость

$$I_{вых} = I_{вх}^{\frac{1}{\gamma}}$$



• Практические задания

- Реализовать полутонирование (dither/error diffusion) для монохромных изображений (результат выводить на экран или в файл)
- Реализовать программу, выполняющую коррекцию цвета в полноцветном изображении путем изменения цветов в разных моделях (рассмотреть RGB и HSV). Результат демонстрировать на примере любого растрового изображения.
- Реализовать сохранение растрового изображения в собственный формат, использующий цветовую модель YCrCb и разные коэффициенты квантования для компонент.

- P. Heckbert, "Color image quantization for frame buffer display," *Computer Graphics*, 16(3), pp. 297-307 (1982).
- R. Ulichney, "Digital Halftoning," *The MIT Press*, 1993.
- R. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," *SID 1975 Symp. Dig. Tech. Papers*, pp. 36-37, 1975.
- B. E. Bayer, "An optimum method for two-level rendition of continuous-tone pictures," *IEEE International Conference on Communications*, vol. 1, pp. 26-11 to 26-15, 1976.