

```
In[359]:= (*Лабораторная работа №5*)  
(*По курсу «Защита информационных процессов в компьютерных системах»*)  
(*Встраивание невидимого ЦВЗ в пространственной области*)
```

```
(*  
    Кутузов Илья  
    А-12М-20  
*)
```

```
ппот = 10;
```

```
In[360]:= (*Задание 2*)  
(*Импортировать контейнер-оригинал в пакет "Mathematica"  
  (Import[]) и определить размеры рисунка-ImageDimensions[.])  
  |импорт |размеры изображения  
im = Import["D:\\GitHub Repos\\stud\\mag\\Sem9\\ЦТЗИ\\Lab5\\Nature_10.jpg"]  
  |импорт |дифференцировать  
ImageDimensions[im]  
  |размеры изображения
```

```
Out[360]=
```



```
Out[361]= {1024, 768}
```

```
In[362]:= (*Задание 3*)
(*Создать массив данных в формате RGB с помощью функции ImageData[***,"Byte"].*)
```

```
id0 = ImageData[im, "Byte"];
Dimensions[id0]
Image[{{{255, 0, 0}}}, "Byte"]
Image[{{{0, 255, 0}}}, "Byte"]
Image[{{{0, 0, 255}}}, "Byte"]
```

```
Out[363]= {768, 1024, 3}
```

```
Out[364]= 
```

```
Out[365]= 
```

```
Out[366]= 
```

In[367]:= (\*Задание 4\*)

(\*4. Провести стирание младшего значащего бита во всех пикселях контейнера-оригинала используя функцию BitClear[\*\*\*,0].\*)

обнулить бит

```
id1 = BitClear[id0, 0];
```

обнулить бит

(\*Задание 5\*)

(\*Создать новое изображение из модифицированных данных-

Image[\*\*\*,"Byte"] и сохранить его (правая кнопка мыши\Save Image as-файл bmp).\*)

изображение байт

сох... изображение

```
im1 = Image[id1, "Byte"]
```

изображение байт

```
im1 == im
```

Out[368]=



Out[369]= False

In[370]:= (\*Задание 6\*)

(\*6. Сформировать серию изображений с 2,3,4,5,6-ю стертыми битами, провести визуальное сравнение изображений и определить порог различимости искажений-выделить и сохранить пример замеченных искажений контейнера-оригинала.\*)

```
st = {};
```

```
Do[AppendTo[st,
```

... добавить в конец к

```
{Image[BitAnd[ImageData[im, "Byte"], BitAnd[BitShiftLeft[255, i]]], "Byte"],
```

изоб... побит... данные изображ... байт

побит... сдвинуть биты влево

байт

```
i]], {i, 0, 8}];
```

```
st
```

Out[372]= { { , 0 } ,



{ , 1 } , { , 2 } ,



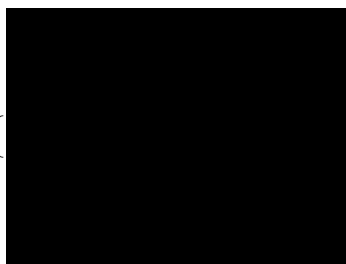
{ , 3 } , { , 4 } ,



{ , 5 } , { , 6 } ,



{ , 7 } , { , 8 } }



In[373]:= ImageData[st[[1]][[1]], "Byte"] == id0

данные изображения байт

ImageData[st[[2]][[1]], "Byte"] == id1

данные изображения байт

Out[373]= True

Out[374]= True



```

In[375]:= (*Задание 7*)
(*Для трех произвольных строк изображения-
оригинала установить в "1" (BitSet[***,k]) все биты в 7,
                                         |установить бит
6 и 5-ом разрядах.Сохранить модифицированное изображение.*)
mask = 2^7 + 2^6 + 2^5;
IntegerDigits[mask, 2]
|цифры целого числа

(*до 240*)
rowcnt = Dimensions[id0][[1]];
|размеры массива
SeedRandom[nnom + 1];
|инициализация генератора псевдослучайных чисел
idrand = id0;
Do[rn = RandomInteger[rowcnt]; idrand[[rn]] = BitOr[idrand[[rn]], mask], {3}];
|опера... |случайное целое число |битное или
Image[idrand, "Byte"]
|изображение |байт

```

```
Out[376]= {1, 1, 1, 0, 0, 0, 0, 0}
```

```
Out[381]=
```



```
In[382]:=
```

```

In[383]:= (*Задание 8*)
(*В строке с номером N изображения-оригинала установить младшие биты зеленого цвета,
   численное приближение
   кратные N в "0", а младшие биты синего цвета, кратные 2N-
   численное приближение численное приближение
   в "1". Сохранить модифицированное изображение.*)
n8 = nnom
idcolor = id0;

Do[idcolor[[n8]][[i]][[3]] = BitSet[idcolor[[n8]][[i]][[3]], 0];,
  оператор цикла установить бит
  {i, 2 * n8, Dimensions[idcolor][[2]], 2 * n8}
  размеры массива
Do[idcolor[[n8]][[i]][[2]] = BitClear[idcolor[[n8]][[i]][[2]], 0];,
  обнулить бит
  {i, n8, Dimensions[idcolor][[2]], n8}
  размеры массива
difs = Tally[id0[[n8]] - idcolor[[n8]]]
  подсчитать
Do[Print[difs[[i]][[1]], " - ",
  печатать
  Position[id0[[n8]] - idcolor[[n8]], difs[[i]][[1]]], {i, 2, Length[difs]}]
  позиция по образцу длина
Image[idcolor, "Byte"]
  байт
Image[Abs[idcolor - id0] * 255, "Byte"]
  абсолютное значение байт

```

Out[383]= 10

```

Out[387]= {{ {0, 0, 0}, 951}, { {0, 1, 0}, 44}, { {0, 1, -1}, 13}, { {0, 0, -1}, 16}}

{0, 1, 0} - {{10}, {40}, {50}, {70}, {90}, {150}, {170}, {210}, {230}, {250}, {270},
  {310}, {350}, {360}, {430}, {440}, {470}, {480}, {500}, {510}, {560}, {570},
  {580}, {590}, {610}, {620}, {630}, {660}, {670}, {690}, {710}, {730}, {760},
  {790}, {810}, {830}, {850}, {890}, {910}, {940}, {950}, {970}, {980}, {990}}

{0, 1, -1} - {{20}, {60}, {100}, {120}, {220}, {260}, {280}, {320}, {340}, {380}, {520}, {880}, {900}}

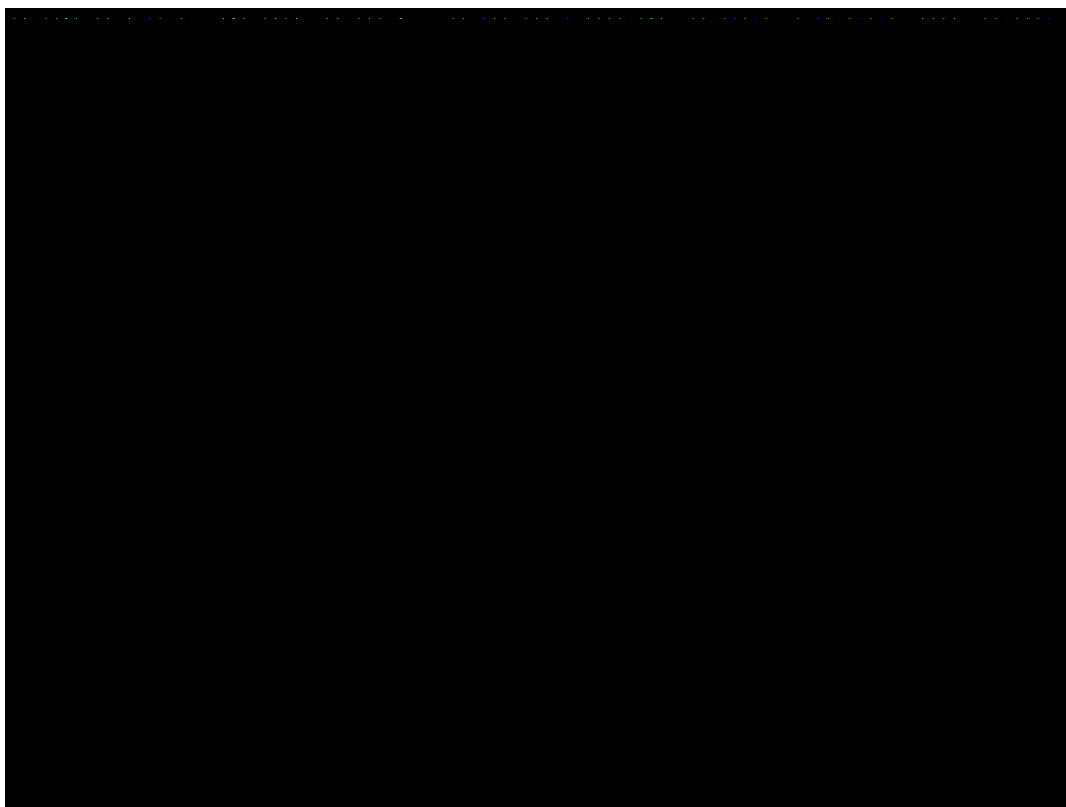
{0, 0, -1} - {{140}, {180}, {200}, {460}, {540}, {600},
  {680}, {700}, {720}, {780}, {800}, {820}, {840}, {920}, {960}, {1000}}

```

Out[389]=



Out[390]=



In[391]:= (\*Задание 9\*)

(\*Сформировать контейнер-оригинал №2:изображение в формате BMP24  
размером 360\*480 (Paint\Рисунок\Атрибуты) из рисунка с номером N,  
находящегося в папке\Lab5-2010\work task\Images 5.\*)

[численное приближение]

```

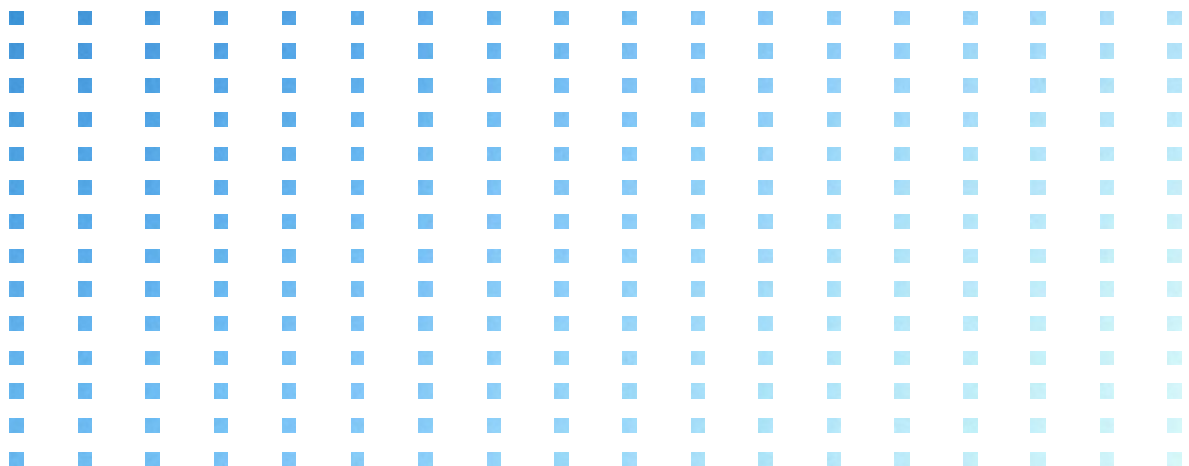
In[392]:= (*Задание 10*)
(*Импортировать контейнер-оригинал №2 в пакет "Mathematica" и разбить его с
помощью функции ImagePartition[***, *] на блоки размером 8x8 пикселей.1*)
ImageDimensions[img9]
img9 = Import["D:\\GitHub Repos\\stud\\mag\\Sem9\\ЦТЗИ\\Lab5\\imgTask9.bmp"]
Grid[imgparts = ImagePartition[img9, {8, 8}]]

```

Out[392]= {360, 480}

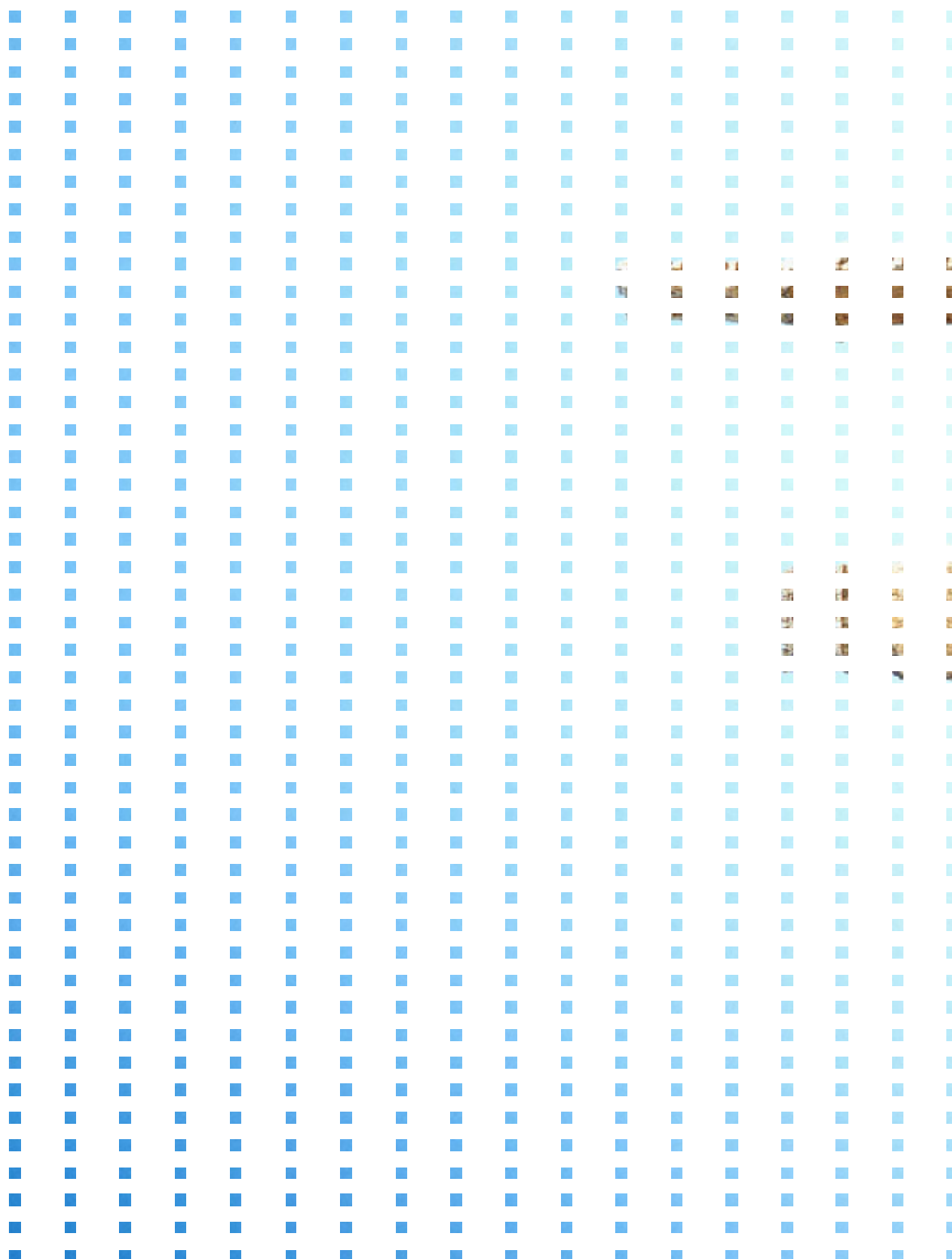


Out[393]=





Out[394]=



In[395]:= (\*Задание 11\*)

(\*1. Определить размерность полученного массива-Dimensions[\*].\*)

|размеры массива

Dimensions[imgparts]

|размеры массива

Out[395]= {60, 45}

```

In[396]:= (*Задание 12*)
n12 = nпом
(*Разделить изображение из блока с индексами 2N,
                                     |численное прибли
2N на три составляющие в помощью функции ColorSeparate[.])
|численное приближение |разделить цветовой кана
img12 = imgparts[[n12 * 2]][[n12 * 2]];
ColorSeparate[img12]
|разделить цветовой каналы

Out[396]= 10

Out[398]= {■, ■, ■}

In[399]:= (*Задание 13*)
(*Для каждого цвета получить массив значений
интенсивностей цвета и выровнять его до уровня списка.*)

In[400]:= colors = {};
Do[AppendTo[colors, ImageData[ColorSeparate[img12][[i]], "Byte"]],
|... |добавить в конец к |данные из... |разделить цветовой каналы |байт
  {i, Length[ColorSeparate[img12][[i]]}
|длина |разделить цветовой каналы
Length[IntegerDigits[Min[colors], 2]] == Length[IntegerDigits[Max[colors], 2]]
|длина |цифры целого ч... |минимум |длина |цифры целого ч... |максимум
colors

Out[402]= True

Out[403]= {{ {225, 225, 228, 229, 229, 231, 231, 231}, {225, 225, 228, 230, 230, 231, 231, 231},
  {226, 226, 227, 228, 229, 230, 231, 231}, {226, 226, 227, 228, 230, 229, 229, 229},
  {225, 225, 227, 228, 229, 230, 231, 231}, {226, 227, 227, 229, 231, 231, 231, 231},
  {228, 229, 228, 229, 231, 230, 229, 229}, {227, 228, 228, 229, 231, 231, 231, 231}},
  {{ {249, 249, 248, 248, 249, 249, 248, 249}, {249, 248, 248, 248, 249, 249, 248, 249},
  {250, 249, 247, 247, 248, 249, 249, 249}, {249, 249, 247, 247, 249, 248, 247, 247},
  {249, 249, 247, 247, 248, 249, 249, 249}, {250, 249, 247, 247, 249, 249, 249, 249},
  {249, 248, 247, 247, 249, 248, 247, 247}, {248, 248, 247, 247, 249, 249, 249, 249}},
  {{ {251, 248, 250, 252, 253, 251, 248, 249}, {250, 248, 250, 253, 253, 251, 248, 249},
  {251, 249, 249, 251, 253, 251, 248, 249}, {251, 249, 249, 251, 253, 250, 247, 247},
  {251, 249, 249, 251, 253, 251, 248, 249}, {251, 249, 249, 251, 253, 251, 248, 249},
  {251, 249, 249, 251, 253, 250, 247, 247}, {250, 248, 249, 251, 253, 251, 248, 249}}}

In[404]:= (*Задание 14*)
(* Определить сумму элементов списка и проверить эту сумму на четность-
Divisible[***,2].*)
|делится на

In[405]:= Do[tot = Total[Flatten[colors[[i]]]];
|оператор... |сумма... |уплотнить
  Print[Divisible[tot, 2], " - ", tot], {i, Length[colors]}]
|печать... |делится на |длина

True - 14 640

True - 15 892

True - 16 000

```

In[406]:=

In[407]:= (\*задание 15\*)

(\*С помощью функции RandomChoice[\*\*\*,\*\*] найти случайные

случайный выбор

индексы для элемента матрицы 8x8\*) rand = RandomChoice[Range[8], 2]

случайный вы... диапазон

Out[407]= {7, 6}

In[408]:=

(\*Задание 16\*)

(\*Изменить четность блока, прибавив по модулю 2 единицу (операция BitXor[\*\*,\*\*])

сложение битов по мс

к значению интенсивности цвета случайного элемента блока. \*)

Do[colors[[i]][[rand[[1]]]][[rand[[2]]]] =

оператор цикла

BitXor[colors[[i]][[rand[[1]]]][[rand[[2]]]], 1], {i, Length[colors]}]

сложение битов по модулю 2длина

Do[tot = Total[Flatten[colors[[i]]]];]

оператор... сумм... упростить

Print[Divisible[tot, 2], " - ", tot], {i, Length[colors]}]

печать... делится надлина

False - 14 641

False - 15 893

False - 16 001

In[410]:= (\*Задание 17\*)

(\*В строке исходного массива блоков (п.10) с номером N

численное приближение

сформировать первые 20 блоков "четных", следующие 20—"нечетные". \*)

row17 = imgparts[[10]]

Out[410]= {

In[411]:= gr17 = {};

Do[AppendTo[gr17, ColorSeparate[row17[[i]]]], {i, Length[row17]}]

... добавить в конец кразделить цветовые каналыдлина

gr17

Out[413]= {

```

In[414]:= r17 = {};
divs = {};
g17 = {};
b17 = {};

Do[AppendTo[r17, ImageData[gr17[[i]][[1]], "Byte"]], {i, Length[gr17]};
... |добавить в конец| данные изображения |байт |длина
Do[AppendTo[g17, ImageData[gr17[[i]][[2]], "Byte"]], {i, Length[gr17]};
... |добавить в конец| данные изображения |байт |длина
Do[AppendTo[b17, ImageData[gr17[[i]][[3]], "Byte"]], {i, Length[gr17]};
... |добавить в конец| данные изображения |байт |длина

Do[tot = Total[Flatten[r17[[i]]]];
|оператор... |сумм... |уплостить
AppendTo[divs, Mod[tot, 2]], {i, Length[gr17]}]
|добавить в конец к |остаток от деления |длина
pd = Partition[divs, 20];
|разбиение на блоки
pd[[2]] = 1 - pd[[2]];
divs = Flatten[pd];
|уплостить
Do[rand17 = RandomChoice[Range[8], 2];
|оператор ци... |случайный вы... |диапазон
r17[[i]][[rand17[[1]]]][[rand17[[2]]]] =
BitXor[r17[[i]][[rand17[[1]]]][[rand17[[2]]]], divs[[i]], {i, 20 * 2}]
|сложение битов по модулю 2
divs2 = {};
Do[tot = Total[Flatten[r17[[i]]]];
|оператор... |сумм... |уплостить
AppendTo[divs2, Mod[tot, 2]], {i, Length[gr17]}]
|добавить в конец к |остаток от деления |длина
divs2

Out[428]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0}

```

In[429]:=



```

In[430]:= is = {};
Do[AppendTo[is, {Image[r17[[i]], "Byte"],
  ...  [добавить в конец] [изображение] [байт]
      Image[g17[[i]], "Byte"], Image[b17[[i]], "Byte"]}], {i, Length[r17]}]
  [изображение] [байт] [изображение] [байт] [длина]
nr17 = {};
Do[AppendTo[nr17, ColorCombine[is[[i]]]], {i, Length[is]}]
  ...  [добавить в конец к] [комбинировать цвета] [длина]
imgparts[[10]] = nr17;

(*Задание 18*)
(*Создать новое изображение из модифицированных данных-
ImageAssemble[]-и сохранить его.*)
[собрать изображение]
newimg = ImageAssemble[imgparts]
  [собрать изображение]
img9 == newimg

```

Out[435]=



Out[436]= False

```

In[437]:= (*Задание 19*)
(*Провести визуализацию встроенной информации (для всего рисунка):
четные блоки закрасить белым цветом, нечетные-черным.*)

```

```

In[438]:= parts2 = ImagePartition[newimg, {8, 8}];
  [разбиение изображения]

```

```

In[439]:= divs19 = {};
Do[
  оператор цикла

  st = parts2[[i]][[j]];
  c19 = ColorSeparate[st];
    разделить цветовые каналы
  r19 = {};
  Do[AppendTo[r19, ImageData[c19[[1]], "Byte"]], Length[c19]];
  ... добавить в коне... данные изображения байт длина
  AppendTo[divs19, Mod[Total[Flatten[r19]], 2]]
    ос... сумм... упростить
  , {i, Dimensions[parts2][[1]]}, {j, Dimensions[parts2][[2]]}}
    размеры массива размеры массива
divs19 = Partition[divs19, Dimensions[parts2][[1]]];
  разбиение на блоки размеры массива
Dimensions[divs19]
  размеры массива

```

Out[442]= {45, 60}

```

In[443]:= Dimensions[ImageData[parts2[[1]][[1]], "Byte"]];
  размеры ма... данные изображения байт

pixel = Partition[Partition[Range[8 * 8 * 3] / Range[8 * 8 * 3], 3], 8];
  разбиение... разбиение... диапазон диапазон
black = Image[pixel * 0, "Byte"]
  изображение байт
white = Image[pixel * 255, "Byte"]
  изображение байт

```

Out[445]=



Out[446]=

```

In[447]:= map = {};
fd = Flatten[divs19];
      |уплостить
Do[AppendTo[map, If[fd[[i]] == 1, black, white]], {i, Length[fd]}]
      |... |добавить в конце... |условный оператор |длина
map = Partition[map, Dimensions[divs19][[1]]];
      |разбиение на бл... |размеры массива
ImageAssemble[map]
      |собрать изображение

```



Out[451]=