

```

Ввод [1]: 1 from PIL import ImageGrab
2 from IPython.display import display, Image
3 def ins(ratio=1.0):
4     im_data = ImageGrab.grabclipboard()
5     new_size = tuple([int(i*ratio) for i in im_data.size])
6     thumb = im_data.resize(new_size)
7     fn = "temp.PNG"
8     thumb.save(fn)
9     img = Image(filename=fn)
10    display(img)

```

```

Ввод [2]: 1 ins(1)

```

**1.23.** (С использованием «Jupyter Notebook» или «Wolfram Mathematica») Используя графические возможности, изобразите матрицу размера  $n \times n$  для  $n = 100$ , каждая строка которой содержит элементы биномиальных коэффициентов по модулю 2 (закрашенный «синий» квадратик соответствует 1, «желтый» закрашенный – 0).

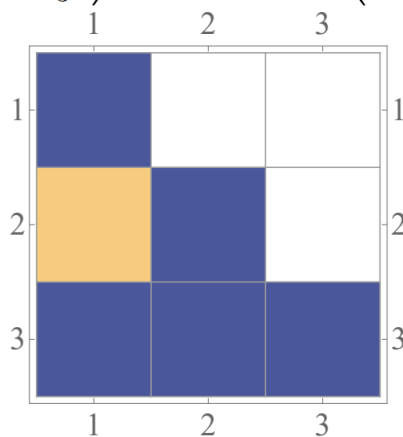
```

Ввод [3]: 1 ins(1)

```

Например, для  $n = 3$  матрица имеет вид

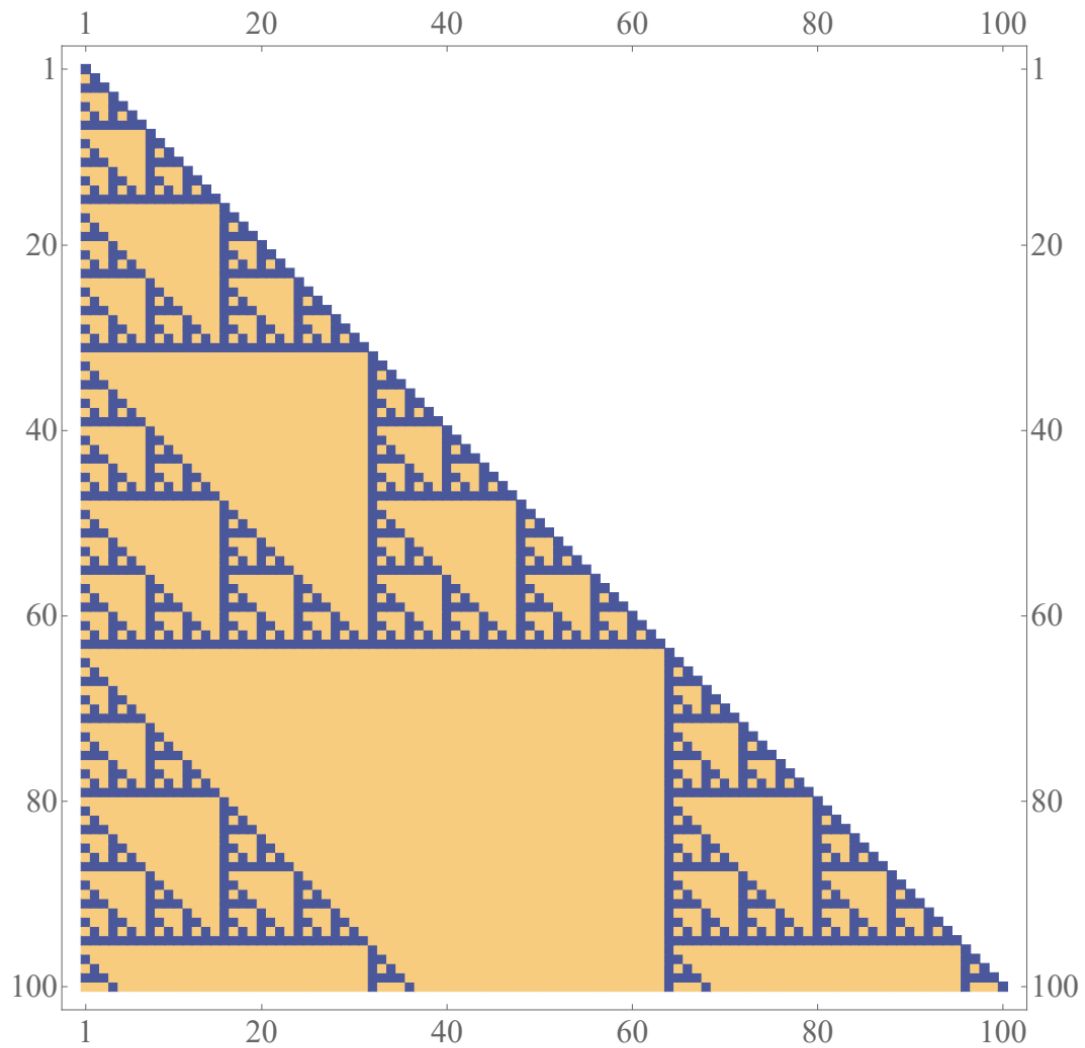
$$\begin{pmatrix} C_1^1 & 0 & 0 \\ C_2^1 & C_2^2 & 0 \\ C_3^1 & C_3^2 & C_3^3 \end{pmatrix} \mod 2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} =$$



Ввод [4]:

```
1 ins(1)
```

**Ответ:** Для  $n = 100$  матрица биномиальных коэффициентов по модулю 2 выглядит так:



Ввод [2]:

```
1 import numpy as np
2 from itertools import *
3 from more_itertools import *
4 from sympy import *
5 from scipy.special import *
6 #####
7 import matplotlib.pyplot as plt
8 import matplotlib as mlt
9 #####
10 import math
11 init_printing()
```

Ввод [3]:

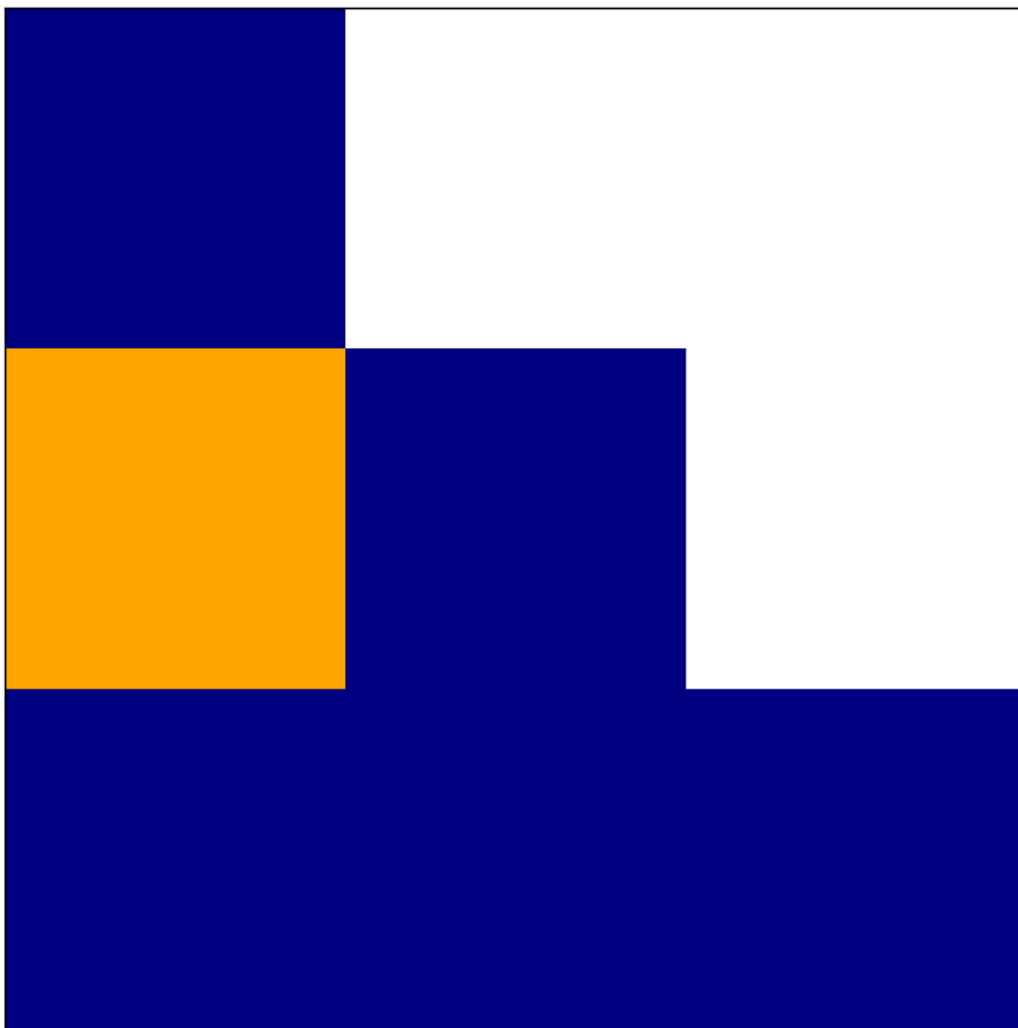
```
1 import matplotlib.pyplot as plt
2 import matplotlib.ticker as ticker
3 from matplotlib import rcParams
4 #####
5 import locale
6 locale.setlocale(locale.LC_NUMERIC, 'russian')
7 plt.rcParams['axes.formatter.use_locale'] = True
8 #####
9 #####
10 plt.rcParams['font.size'] = 16
11 plt.rcParams["font.family"] = "Times New Roman"
12 plt.rcParams['mathtext.fontset'] = 'cm'
13 #####
```

Ввод [6]:

```
1 def get_matrix(n):
2     return ([[math.comb(i, j) % 2 if i>=j else -1 for j in range(1, n + 1)]
3             for i in range(1, n + 1)])
4 col = ['w', 'orange', 'navy']
5
```

Ввод [14]:

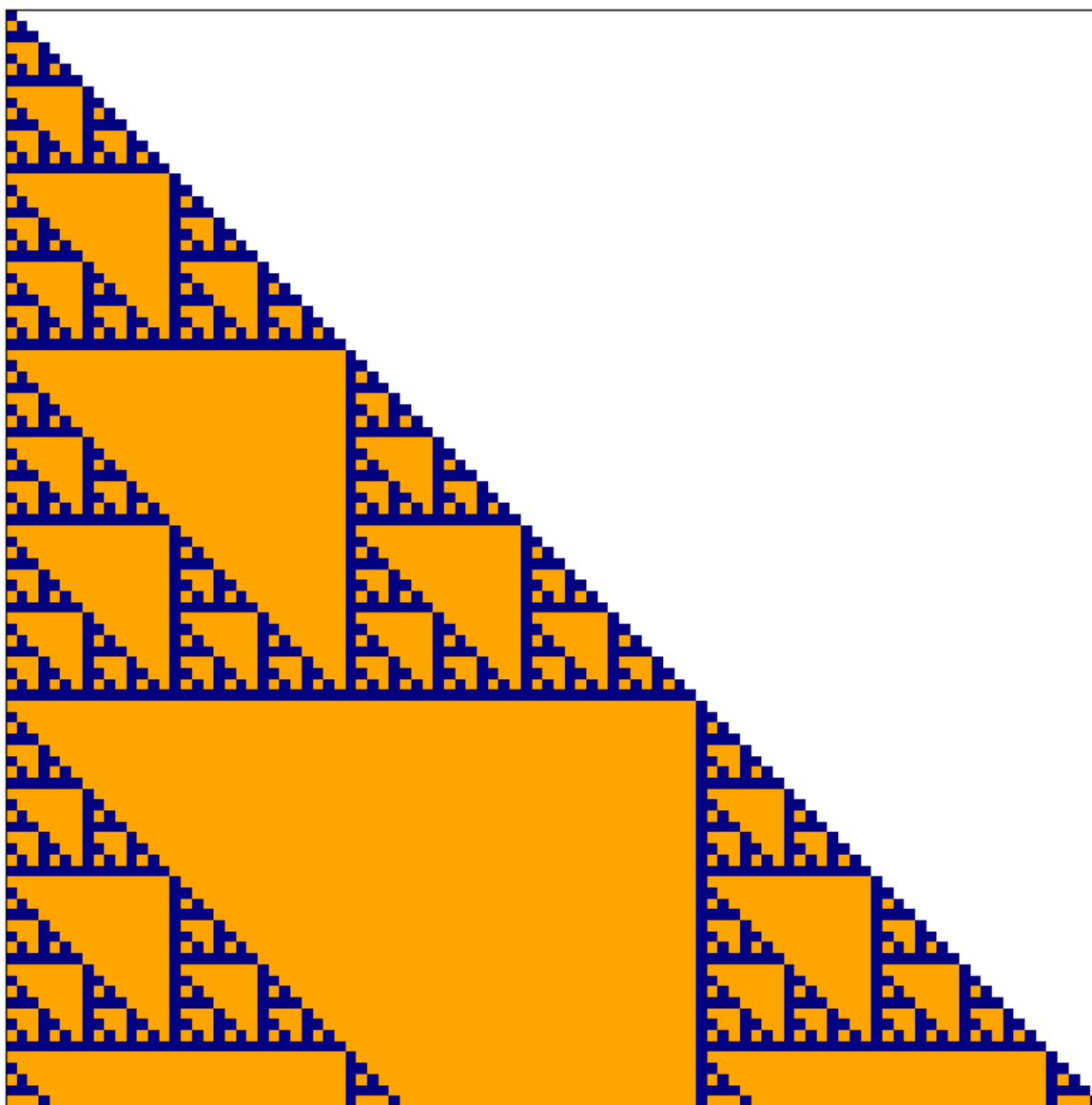
```
1 fig,ax =plt.subplots(figsize=(8, 8))
2 plt.tick_params(labelsize = 20)
3 #plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=0.5)
4 ax.xaxis.set_major_locator(ticker.MaxNLocator(6))
5 ax.yaxis.set_major_locator(ticker.MaxNLocator(6))
6 plt.tick_params(left = False, right = False , labelleft = False, labelbottom = False, bot
7 #####
8
9 n = 3
10
11 #plt.figure(figsize = (12, 12))
12 plt.imshow(get_matrix(n), cmap = mlt.colors.LinearSegmentedColormap.from_list('myowncmap
13 plt.show()
14 fig.savefig("sol_1_23a.pdf", bbox_inches='tight')
```



```

Ввод [13]: 1 fig,ax =plt.subplots(figsize=(10, 10))
2 plt.tick_params(labelsize = 20)
3 #plt.grid(color='DarkTurquoise', alpha=0.75, linestyle=':', linewidth=0.2)
4 ax.xaxis.set_major_locator(ticker.MaxNLocator(6))
5 ax.yaxis.set_major_locator(ticker.MaxNLocator(6))
6 plt.tick_params(left = False, right = False , labelleft = False, labelbottom = False, bot
7 #####
8
9 n = 100
10
11 #plt.figure(figsize = (12, 12))
12 plt.imshow(get_matrix(n), cmap = mlt.colors.LinearSegmentedColormap.from_list('myowncmap
13 plt.show()
14 fig.savefig("sol_1_23b.pdf", bbox_inches='tight')

```



```

Ввод [9]: 1 Matrix(get_matrix(3))

```

```

Out[9]: 
$$\begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$


```

```

Ввод [ ]: 1

```

