```python
In [2]:  from IPython.display import Image
         Image(filename='screenshot.png')
```
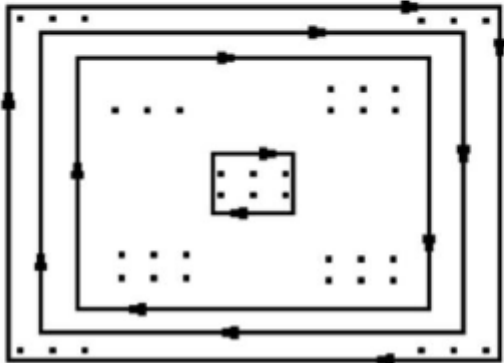
Out[2]:
Вариант 14

Осуществить циклический сдвиг элементов матрицы размером M x N (M строк x N столбцов). Сдвиг выполнить вправо на k элементов таким образом: элементы первой строки сдвигаются в последний столбец сверху вниз, из него - в последнюю строку справа налево, из нее - в первый столбец снизу вверх, из него - в первую строку; для остальных элементов - аналогично.


```python
In [3]:  import numpy as np
```

```python
In [4]:  def full_circle_shift_clockwise(matrix_array: np.array, index_of_circle: int):
             not_shifted = matrix_array.copy()

             # init indexes
             last_index = 1

             # shift first row (init first and slice other)
             matrix_array[index_of_circle, index_of_circle] = not_shifted[index_of_circle + 1, index_of_circle]

             for i in range(index_of_circle + 1, matrix_array.shape[1] - index_of_circle):
                 matrix_array[index_of_circle, i] = not_shifted[index_of_circle, i - 1]

             # shift last column
             for i in range(index_of_circle + 1, matrix_array.shape[0] - index_of_circle):
                 matrix_array[i, matrix_array.shape[1] - index_of_circle - 1] = not_shifted[
                     i - 1, matrix_array.shape[1] - index_of_circle - 1]

             # shift last row
             for i in reversed(range(index_of_circle, matrix_array.shape[1] - index_of_circle - 1)):
                 matrix_array[matrix_array.shape[0] - index_of_circle - 1, i] = not_shifted[
                     matrix_array.shape[0] - index_of_circle - 1, i + 1]

             for i in reversed(range(index_of_circle + 1, matrix_array.shape[0] - 1 - index_of_circle)):
                 matrix_array[i, index_of_circle] = not_shifted[i + 1, index_of_circle]
```

```python
In [5]:  def get_matrix_shift(matrix_array: np.array, shift_times: int = 1):
             result = matrix_array.copy()

             for _ in range(0, shift_times):
                 for shift_index in range(0, int(np.min(matrix_array.shape) / 2)):
                     full_circle_shift_clockwise(result, shift_index)

             return result
```

```python
In [6]:  matrix = np.arange(0, 8 * 10).reshape(8, 10)
```

```python
In [7]:  print(matrix)
```
```
[[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]
 [30 31 32 33 34 35 36 37 38 39]
 [40 41 42 43 44 45 46 47 48 49]
 [50 51 52 53 54 55 56 57 58 59]
 [60 61 62 63 64 65 66 67 68 69]
 [70 71 72 73 74 75 76 77 78 79]]
```

```python
In [8]:  print(get_matrix_shift(matrix))
```
```
[[10  0  1  2  3  4  5  6  7  8]
 [20 21 11 12 13 14 15 16 17  9]
 [30 31 32 22 23 24 25 26 18 19]
 [40 41 42 43 33 34 35 27 28 29]
 [50 51 52 44 45 46 36 37 38 39]
 [60 61 53 54 55 56 57 47 48 49]
 [70 62 63 64 65 66 67 68 58 59]
 [71 72 73 74 75 76 77 78 79 69]]
```

```python
In [9]:  matrix = np.arange(0, 10 * 10).reshape(10, 10)
```

```python
In [10]:  print(matrix)
```
```
[[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]
 [30 31 32 33 34 35 36 37 38 39]
 [40 41 42 43 44 45 46 47 48 49]
 [50 51 52 53 54 55 56 57 58 59]
 [60 61 62 63 64 65 66 67 68 69]
 [70 71 72 73 74 75 76 77 78 79]
 [80 81 82 83 84 85 86 87 88 89]
 [90 91 92 93 94 95 96 97 98 99]]
```

```python
In [11]:  print(get_matrix_shift(matrix))
```
```
[[10  0  1  2  3  4  5  6  7  8]
 [20 21 11 12 13 14 15 16 17  9]
 [30 31 32 22 23 24 25 26 18 19]
 [40 41 42 43 33 34 35 27 28 29]
 [50 51 52 53 54 44 36 37 38 39]
 [60 61 62 63 55 45 46 47 48 49]
 [70 71 72 64 65 66 56 57 58 59]
 [80 81 73 74 75 76 77 67 68 69]
 [90 82 83 84 85 86 87 88 78 79]
 [91 92 93 94 95 96 97 98 99 89]]
```

```python
In [12]:  matrix = np.arange(0, 4 * 10).reshape(4, 10)
```

```python
In [13]:  print(matrix)
```
```
[[ 0  1  2  3  4  5  6  7  8  9]
 [10 11 12 13 14 15 16 17 18 19]
 [20 21 22 23 24 25 26 27 28 29]
 [30 31 32 33 34 35 36 37 38 39]]
```

```python
In [14]:  print(get_matrix_shift(matrix))
```
```
[[10  0  1  2  3  4  5  6  7  8]
 [20 21 11 12 13 14 15 16 17  9]
 [30 22 23 24 25 26 27 28 18 19]
 [31 32 33 34 35 36 37 38 39 29]]
```

```python
In [15]:  print(get_matrix_shift(matrix, 3))
```
```
[[30 20 10  0  1  2  3  4  5  6]
 [31 23 22 21 11 12 13 14 15  7]
 [32 24 25 26 27 28 18 17 16  8]
 [33 34 35 36 37 38 39 29 19  9]]
```

```python
In [16]:  print(get_matrix_shift(matrix, 12))
```
```
[[39 38 37 36 35 34 33 32 31 30]
 [29 15 16 17 18 28 27 26 25 20]
 [19 14 13 12 11 21 22 23 24 10]
 [ 9  8  7  6  5  4  3  2  1  0]]
```

```python
In [17]:  matrix = np.arange(0, 10 * 8).reshape(8, 10)
```

```python
In [18]:  print(get_matrix_shift(matrix, 3))
```
```
[[30 20 10  0  1  2  3  4  5  6]
 [40 41 31 21 11 12 13 14 15  7]
 [50 51 52 42 32 22 23 24 16  8]
 [60 61 53 45 44 43 33 25 17  9]
 [70 62 54 46 36 35 34 26 18 19]
 [71 63 55 56 57 47 37 27 28 29]
 [72 64 65 66 67 68 58 48 38 39]
 [73 74 75 76 77 78 79 69 59 49]]
```

```python
In [ ]:
```