

Методы

№10

Задача 4

import random

(10)

Примеры не проверяются

def get_arr(x, y):

```

    return [[0 for i in range(y)] for i in range(x)]
matrix = get_arr(64, 64)

```

def get_pixel_matrix(matrix):

pixel_matrix = get_arr(8, 8)

```

    for i in range(64):

```

```

        for j in range(64):

```

```

            pixel_matrix[i//8][j//8] += matrix[i][j]

```

```

    for i in range(8):

```

```

        for j in range(8):

```

```

            pixel_matrix[i][j] = int(pixel_matrix[i][j])

```

```

    return pixel_matrix

```

def set_pixel(matrix, x, y, value = 1):

```

    for i in range(x, x+8):

```

```

        for j in range(y, y+8):

```

```

            matrix[i][j] = value

```

def set_big_pixel(matrix, x, y, value = 1):

```

    set_pixel(matrix, x*8, y*8, value)

```

def sum_pixel(matrix, x, y):

s = 0

```

    for i in range(x, x+8):

```

```

        for j in range(y, y+8):

```

```

            if i >= 64 or j >= 64:

```

```

                s += 1

```

```

            else:

```

```

                s += matrix[i][j]

```

```

    return s

```

```
([0]x : x unique) = dot (sum(x)) zeros = sum(x) do ...
```

Figure

```
def sum = figure (matrix, x, y):  
    s = 0  
    s += sum - pixel (matrix, x, y)  
    s += sum - pixel (matrix, x+1, y)  
    s += sum - pixel (matrix, x+1, y+1)  
    s += sum - pixel (matrix, x+1, y+2)  
    return s
```

```
def set_figure (matrix, x, y, value = 1):  
    set_pixel (matrix, x, y, value)  
    set_pixel (matrix, x+1, y, value)  
    set_pixel (matrix, x+1, y+1, value)  
    set_pixel (matrix, x+1, y+2, value)
```

distance = 8

figures = []

while True:

a = input()

if not a:

break

if a == "add":

pixel_matrix = get_pixel_matrix (matrix)

available_positions = []

for i in range (7):

for j in range (6):

if pixel_matrix[i][j] == 0 and pixel_matrix[i+1][j] == 0 and pixel_matrix[i+1][j+1] == 0 and pixel_matrix[i+1][j+2] == 0:

available_positions.append ((i, j))

if len (available_positions) == 0:

print ("no more moves possible")
continue

pos = random.choice (available_positions)
x = pos[0] * 8

y = pos[1] * 8

np.random.choice

else:

next_ind = indexes[-1] + 1

set_figure(matrix, x, y, 1)

Безопасно 4

figures.append([x, y])

print("quyppa qolabrenna")

elif a == "distance":

print("elguziz qusangusar repensuynus quyppan
za ogro geisibue")

distance = int((int(input()) * 8) / 100)

print(f"repensuynus za ogro geisibue oguzgeisibue-
es na {int(distance / 8)} pr")

elif a == "show":

pixel_matrix = get_pixel_matrix(matrix)

for i in range(8):

print(' '.join(map(str, pixel_matrix[i])))

elif a == "down":

figures = sorted(figures, key=lambda x: -x[0])

for figure in figures:

for i in range(distance):

set_figure(matrix, figure[0], figure[1], 0)

if sum_figure(matrix, figure[0] + 1, figure[1]) !=
set_figure(matrix, figure[0], figure[1], 1)
break

else:

figure[0] += 1

set_figure(matrix, figures[0], figure[1], 1)

print("quyppan oguzgeisibue esna")

elif a == "up":

figures = sorted(figures, key=lambda x: x[0])

for figure in figures:

for i in range(distance):

BAFUNDANT 4

```
if figure[0] == 0:
    break
```

```
set_figure(matrix, figure[0], figure[1], 0)
```

```
if sum_figure(matrix, figure[0]-1, figure[1])>0:
```

```
    set_figure(matrix, figure[0], figure[1], 1)
    break
```

```
else:
```

```
    figure[0] -= 1
```

```
    set_figure(matrix, figure[0], figure[1], 1)
```

```
print("quraypa cyburymas lepx")
```

```
elif a == "left":
```

```
    figures = sorted(figures, key = lambda x: x[1])
```

```
    for figure in figures:
```

```
        for i in range(distance):
```

```
            if figure[1] == 0:
```

```
                break
```

```
            set_figure(matrix, figure[0], figure[1], 0)
```

```
            if sum_figure(matrix, figure[0], figure[1]-1)>0:
```

```
                set_figure(matrix, figure[0], figure[1], 1)
```

```
                break
```

```
            else:
```

```
                figure[1] -= 1
```

```
                set_figure(matrix, figure[0], figure[1], 1)
```

```
print("quraypa cyburymas lebo")
```

```
elif a == "right":
```

```
    figures = sorted(figures, key = lambda x: x[1])
```

```
    for figure in figures:
```

```
        for i in range(distance)
```

```
            set_figure(matrix, figure[0], figure[1], 0)
```

Copyright 4

```
sum = figure(matrix, figure[0], figure[1] + 1)
set_figure(matrix, figure[0], figure[1], 1)
break
```

else:

```
figure[1] += 1
```

```
set_figure(matrix, figure[0], figure[1], 1)
```

```
print("anyway cycles are done")
```

~ 9 (20)

```
indexes = []
```

```
arr_size = 256
```

```
arr = [0 for i in range(arr_size)]
```

```
cycles = [0 for i in range(arr_size)]
```

```
max_cycles = 125
```

```
while True:
```

```
    a = input()
```

```
    if not a:
```

```
        break
```

```
    if a == "cycles":
```

```
        print("please enter a number")
```

```
        numb = int(input())
```

```
        print(f"area {numb} area is {arr[numb]}")
```

```
    elif a == "all":
```

```
        for index in indexes:
```

```
            time = (index + 1) % arr_size
```

```
            T = (index + 2) % arr_size
```

```
            print(f"[arr[index]] : {arr[time]} T={arr[T]-50}")
```

```
    elif a == "add":
```

```
        print("please enter a value")
```

```
        hour = int(input())
```

```
        print("please enter a number")
```

```
        minutes = int(input())
```

Вариант 1

print ("Введите количество элементов массива")

T = int(input()) + 50

if len(indexes) == 0:

next_ind = 0

else:

next_ind = indexes[-1] + 3

if next_ind + 3 >= indexes[0]:

indexes = indexes[1:]

indexes.append(next_ind)

time = (next_ind + 1) % arr_size

T_ind = (next_ind + 2) % arr_size

arr[next_ind] = hour

arr[time] = minutes

arr[T_ind] = T

cycles[next_ind] += 1

cycles[time] += 1

cycles[T_ind] += 1

elif a == "test":

count = int((arr_size * (max_cycles) -
sum(cycles)) / 3)

print ("В промежуток времени между
записей еще {count} записей")

н 7 ~~1/6~~

Счет: 2000.

Обоснование: Внешний цикл означает, что мы
итерируем пока последнее число
в массиве больше 40. Во внутреннем
цикле мы делаем swap соседних
элементов пока не дойдем до
последнего числа деленного на 2,
так как на каждой итерации
последнее число точно swapнется на
большее, то итераций будет 40.
Теперь посмотрим сколько итераций
будет делать внутренний цикл.

Вариант 4

на первой итерации внешнего цикла
сделает 40 итераций - дойдет от 80
до 40. После этого последнее число
уменьшится на 1 и внутренний цикл
станет на 1 итерацию больше. Так
как последнее число каждый раз
уменьшается ~~на~~ на 1, а цикл идет
~~пока~~ пока $a[i-1] > x = (\text{последнее число} / 2)$,
то количество итераций внутреннего цикла
будет увеличиваться на 1 каждые 2 ите-
рации внешнего. Итого имеем $40 + 41 * 2 +$
 $+ 59 * 2 + 60 = 2000$.