

# Сегодня в уроке

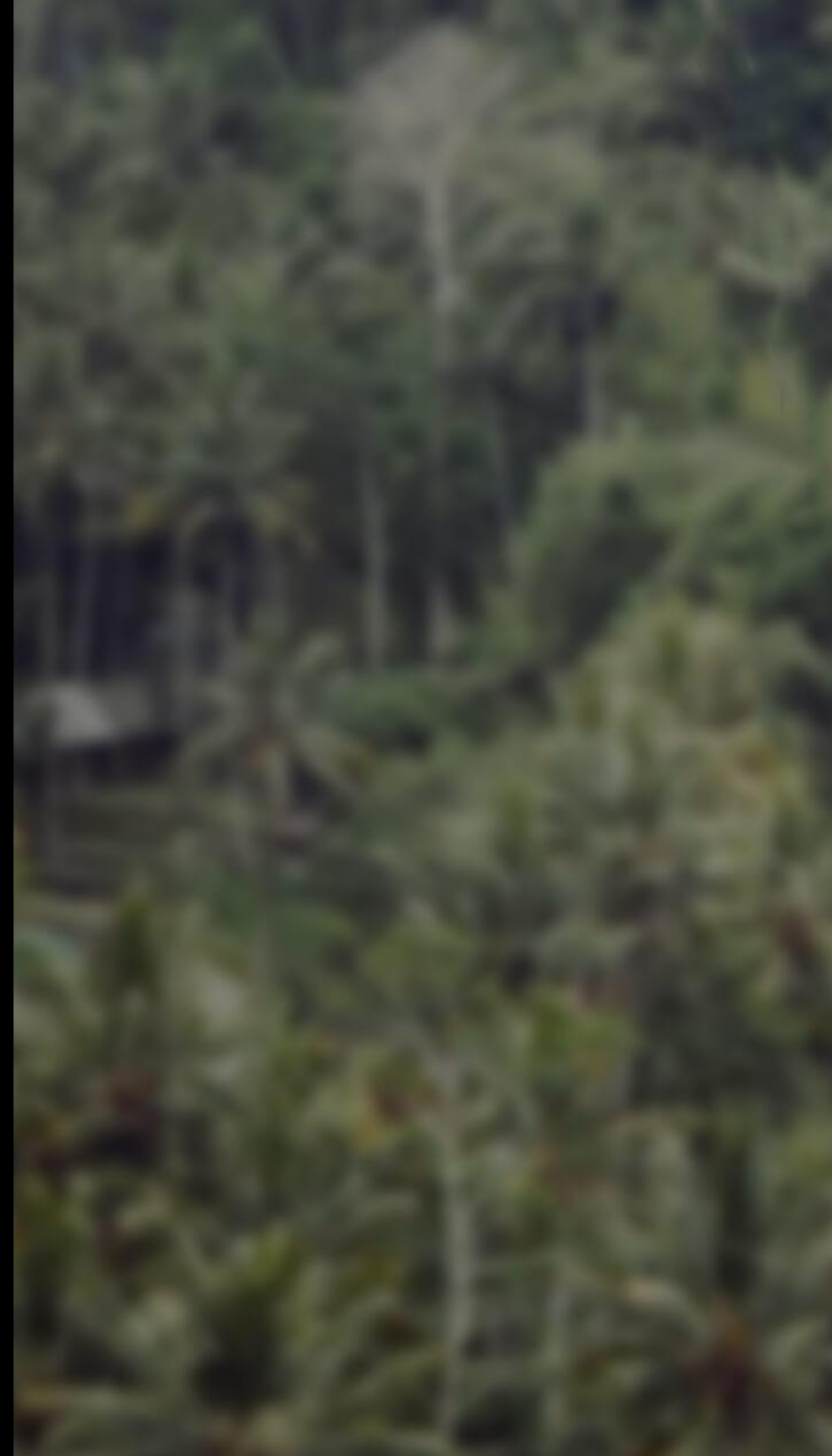
- Задание 25

**ВЕРОНИКА  
ПАВЛОВНА**

Геймификация  
подготовки к ЕГЭ  
по информатике



pavlovna\_ege



# Проверяемый элемент ФГОС

Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления:

- разбиение записи числа на отдельные цифры
- нахождение суммы и произведения цифр
- нахождение максимальной (минимальной) цифры
- представление числа в виде набора простых сомножителей
- алгоритм быстрого возведения в степень
- поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена»

# Проверяемый элемент

- разбиение записи числа на отдельные цифры (в 10 с.с и в стандартных 2 способа, в N через цикл while)

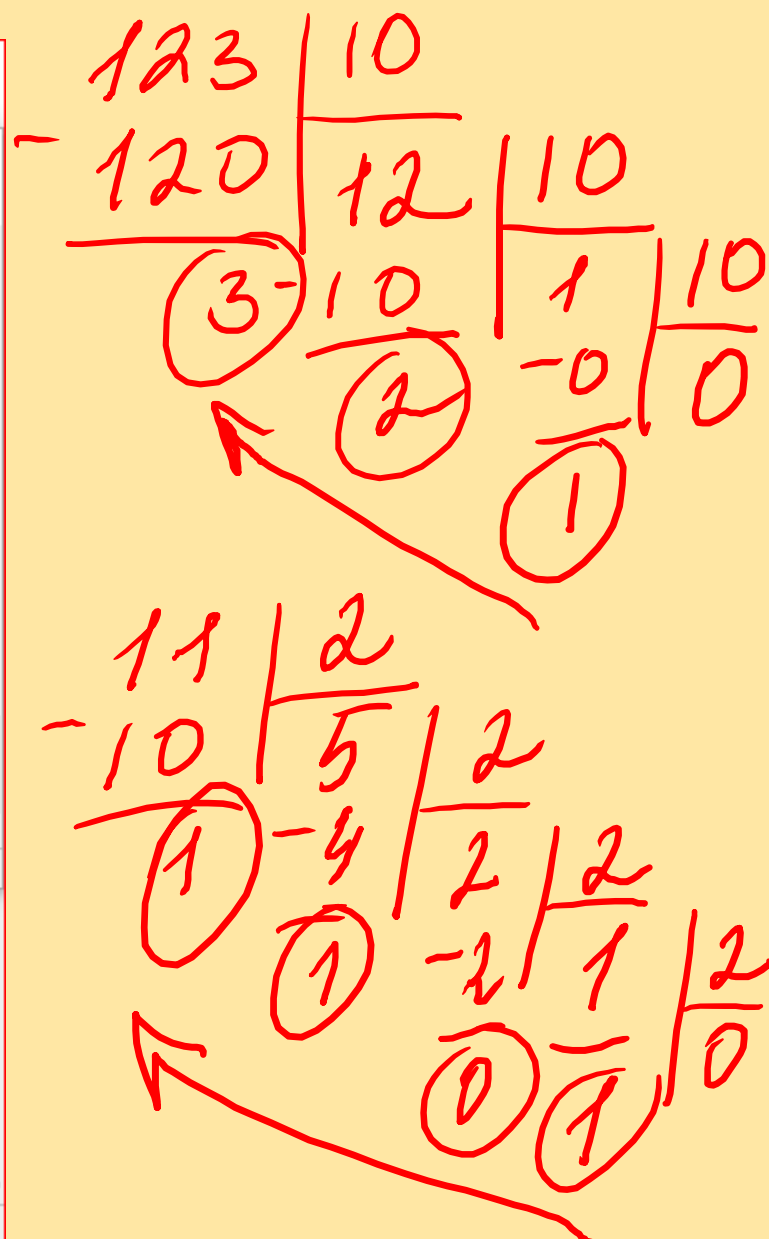
```
*000.py - D:\000.py (3.11.3)*
File Edit Format Run Options Window Help
# 10 с.с. через цифры
N = 123
while N != 0:
    print(N % 10)
    N //= 10
# 10 с.с через строки
n = str(123)
print([int(i) for i in n])

# 5 с.с.
N = 123
while N != 0:
    print(N % 5)
    N //= 5

# 2, 8, 16
N = 123
print(bin(N) [2:], oct(N) [2:], hex(N) [2:])

IDLE Shell 3.11.3
File Edit Shell Debug Options Window Help
D:\000.py =====
=
3
2
1
[1, 2, 3]
3
4
4
1111011 173 7b
>>>
Ln: 66 Col: 0

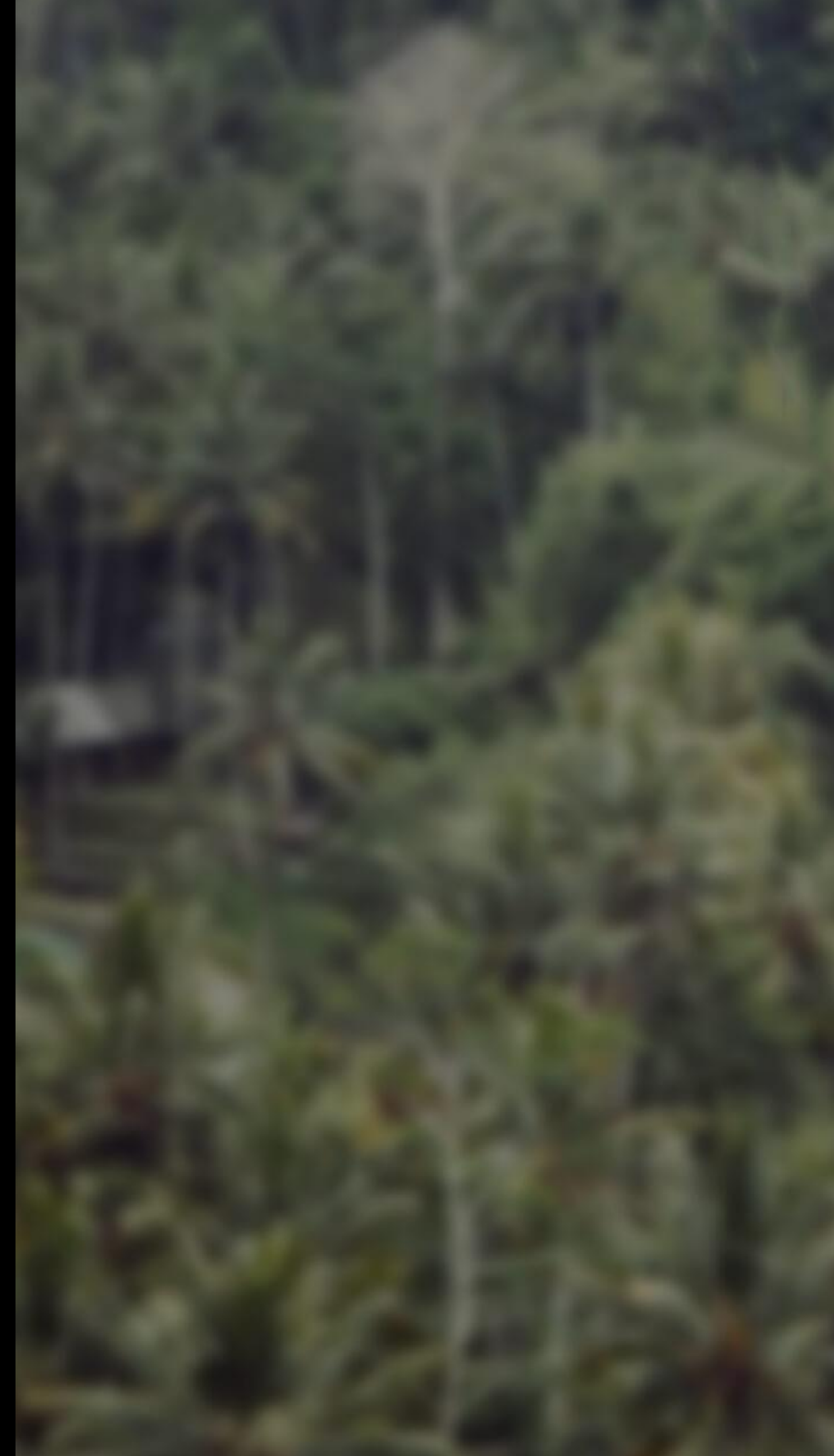
Ln: 1 Col: 21
```





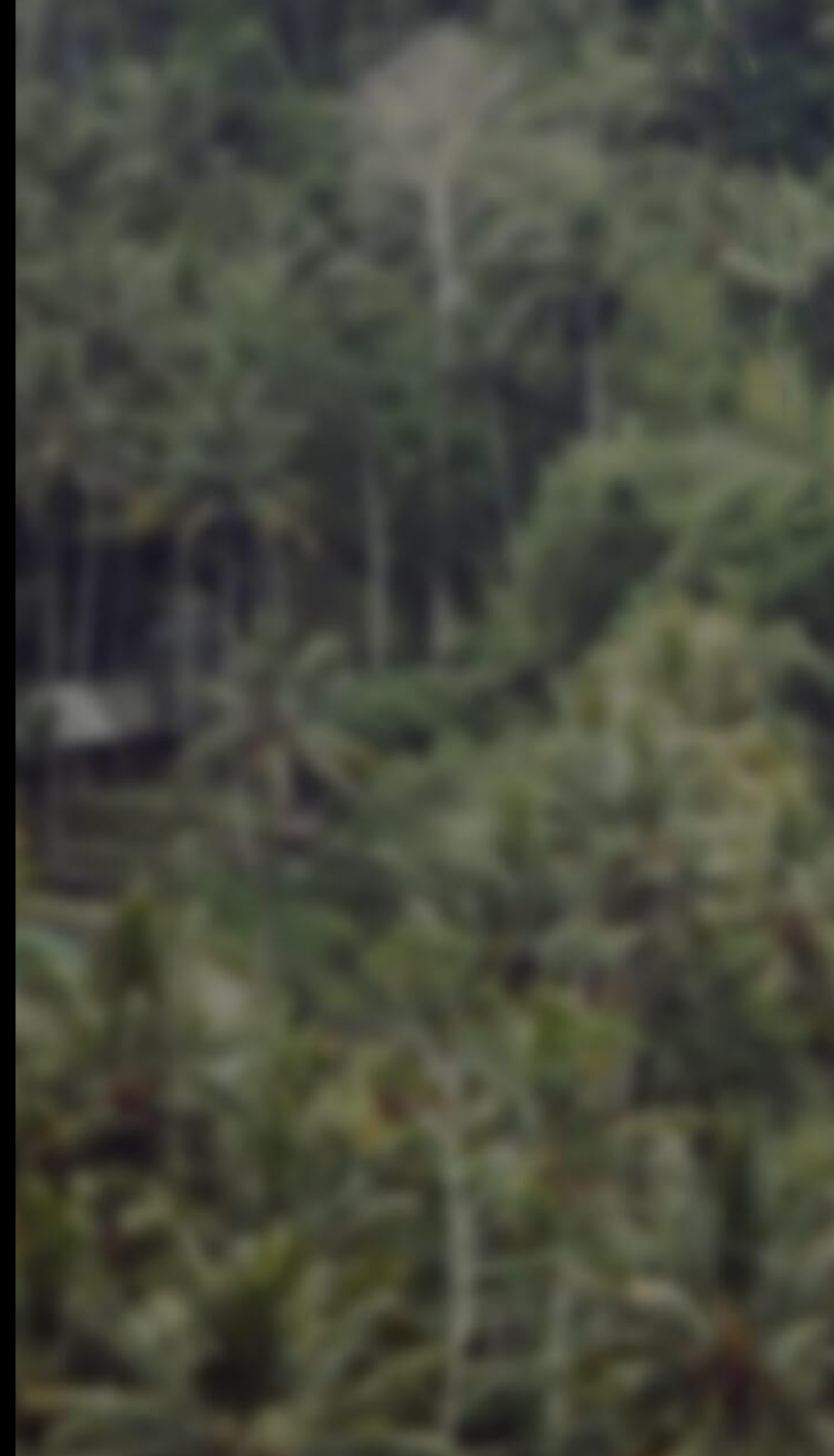
# Проверяемый элемент

- нахождение суммы и произведения цифр



# Проверяемый элемент

- нахождение максимальной (минимальной) цифры



# Проверяемый элемент

- представление числа в виде набора простых сомножителей

Основная теорема арифметики:

Любое число единственным способом представляется в виде произведения простых чисел:

$$n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$$

Число нетривиальных делителей:

$$\delta = (k_1 + 1)(k_2 + 1) \dots (k_m + 1) - 2$$

Если  $\delta = 3$ :

$$(k_1 + 1)(k_2 + 1) \dots (k_m + 1) = 5$$

$$k_1 = 4, k_2 = k_3 = \dots k_m = 0$$

```
def Factor(n):  
    Ans = []  
    d = 2  
    while d * d <= n:  
        if n % d == 0:  
            Ans.append(d)  
            n //= d  
        else:  
            d += 1  
    if n > 1:  
        Ans.append(n)  
    return Ans  
  
print(Factor(30))
```

```
File Edit Format Run Options Window Help  
N = 20  
d = 2  
while N != 1:  
    while N % d == 0:  
        print(d)  
        N //= d  
    d += 1
```

$$\begin{array}{r|l} 20 & 2 \\ 10 & 2 \\ 5 & 5 \\ 1 & \end{array}$$

$$2^2 \cdot 5^1$$
$$(2+1)(1+1) = 3 \cdot 2 = 6$$
$$1 \ 2 \ 4 \ 5 \ 10 \ 20$$



# Проверяемый элемент

- алгоритм быстрого возведения в степень

```
>>> 5**2  
25
```

```
>>> import math  
>>> math.pow(5, 2)  
25.0
```

```
def fast(base, exponent):  
    result = 1  
    while exponent > 0:  
        if exponent % 2 == 1:  
            result *= base  
        base *= base  
        exponent //= 2  
    return result  
  
print(fast(2, 5))
```

## Проверяемый элемент

- поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена»

```
File Edit Format Run Options Window Help
N = 10**6
d = {i:True for i in range(2, N+1)}
for i in range(2,int((N+1)**0.5)):
    if d[i]:
        for i in range(i*i, N+1, i):
            d[i] = False
d1 = [i for i in range(2,len(d)) if d[i]]
print(len(d1))
```

{ 2: True,  
3: True,  
4: ~~True~~, False

1... 20

1	2	4	5	10	20
---	---	---	---	----	----

A diagram illustrating a sequence of numbers: 1, 2, 4, 5, 10, 20. A vertical line separates the first three numbers (1, 2, 4) from the last three (5, 10, 20). A horizontal line is drawn below the first three numbers, and another horizontal line is drawn below the last three numbers. A large curved line connects the bottom of the first three numbers to the bottom of the last three numbers, passing under the vertical line.

$2^2$   $3^2$   $5^2$   


---

 2 3 ~~4~~ 5 ~~6~~ 7  
~~8~~ ~~9~~ ~~10~~ 11 ~~12~~  
 13 ~~14~~ ~~15~~

2 3 5

$$1..[1..N]$$
$$2. [2; N/2]$$

3.  $[1; N^{0.5}]$   $\text{range}(1, \text{int}(N^{0.5}) + 1)$



## Задание 25.1. Способ 1. Числа + срезы

Маска числа — это последовательность цифр, в которой могут встречаться специальные символы «?» и «\*». Символ «?» означает ровно одну произвольную цифру, символ «\*» означает произвольную (в том числе пустую) последовательность цифр.

Например, маске 123\*4?5 соответствуют числа 123405 и 12376415. Найдите все натуральные числа, не превышающие  $10^{10}$ , которые соответствуют маске  $1*4239?7$  и при этом без остатка делятся на 3147. В ответе запишите все найденные числа в порядке возрастания.

```
File Edit Format Run Options Window Help
for i in range(3147, 10**10+1, 3147):
    s = str(i)
    if s[0]=='1' and s[-6:-2] == '4239' and s[-1] == '7':
        print(i)

IDLE Shell 3.11.3
File Edit Shell Debug Options Window Help
----- RESTART: D:\0000.py -----
100423917
176423967
1374423927
1450423977
1973423907
>>>
```

## Задание 25.1. Способ 2. Комбинации

Маска числа — это последовательность цифр, в которой могут встречаться специальные символы «?» и «\*». Символ «?» означает ровно одну произвольную цифру, символ «\*» означает произвольную (в том числе пустую) последовательность цифр.

Например, маске 123\*4?5 соответствуют числа 123405 и 12376415. Найдите все натуральные числа, не превышающие 10 в 10, которые соответствуют маске 1\*4239?7 и при этом без остатка делятся на 3147. В ответе запишите все найденные числа в порядке возрастания.

```
File Edit Format Run Options Window Help
#10 000 000 000
# 1 *** 423 9?7
k = ['', '00', '000'] + [str(i) for i in range(1000)]
for i in k:
    for j in '0123456789':
        s = f'1{i}4239{j}7'
        if int(s) % 3147 == 0:
            print(s, int(s)//3147)
```

## Задание 25.1. Способ 3. fnmatch

Маска числа — это последовательность цифр, в которой могут встречаться специальные символы «?» и «\*». Символ «?» означает ровно одну произвольную цифру, символ «\*» означает произвольную (в том числе пустую) последовательность цифр.

Например, маске 123\*4?5 соответствуют числа 123405 и 12376415. Найдите все натуральные числа, не превышающие 10 в 10, которые соответствуют маске 1\*4239?7 и при этом без остатка делятся на 3147. В ответе запишите все найденные числа в порядке возрастания.

```
File Edit Format Run Options Window Help
from fnmatch import *
for x in range(0, 10**10, 3147):
    if fnmatch(str(x), '1*4239?7'):
        print(x)
```



## Задание 25.2

Найдите все натуральные числа, не превосходящие 10 в 9, для которых выполнены все условия:

- соответствуют маске  $*31*65?$ ;
- делятся на 31 и 2031 без остатка;
- количество делителей числа является результатом любой степени двойки.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, справа от каждого числа их частное от деления на 2031.