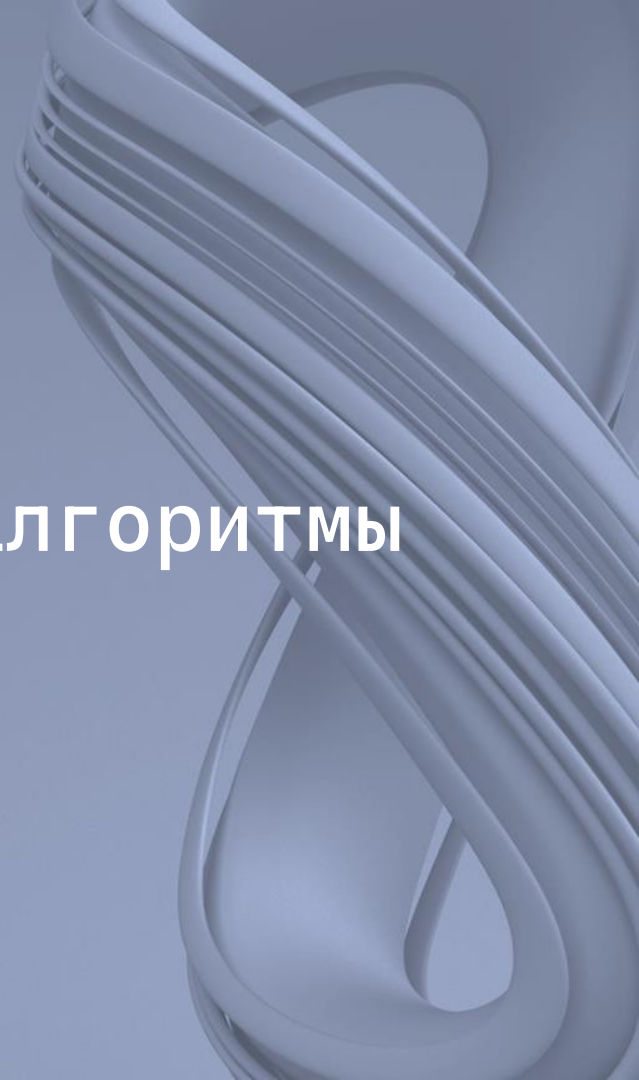


# Модуль #5

## Системы счисления и алгоритмы



- В данном модуле изучим задания КЕГЭ #5 (анализ алгоритмов для чисел) и #14 (операции в системах счисления).

Система счисления – способ записи чисел цифрами и (или) символами из некоторого конечного алфавита.

Алфавит системы счисления – набор цифр и букв (символов), используемых для представления числа.

Основание системы счисления – количество символов, используемых для записи числа (число символов в алфавите).

10-ая СС	2-ая СС	8-ая СС	16-ая СС
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

# Перевод из любой системы счисления в десятичную

Для перевода числа из заданной СС в десятичную необходимо:

1. Пронумеровать разряды числа справа налево (от младшего), начиная с нулевого
2. Вычислить сумму произведения цифр числа на основание системы счисления в степени, соответствующей позиции цифры в числе (из п. 1)

Примеры:

$$1101_2 = 1 * 2^0 + 0 * 2^1 + 1 * 2^2 + 1 * 2^3 = 1 + 4 + 8 = 13_{10}$$

$$17F_{16} = 15 * 16^0 + 7 * 16^1 + 1 * 16^2 = 15 + 112 + 256 = 383_{10}$$

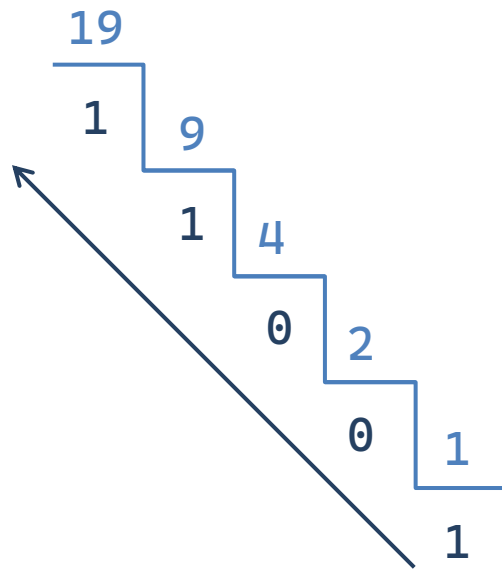
# Перевод из десятичной в любую систему

Для перевода целого десятичного числа в систему счисления с любым основанием необходимо:

1. Последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления до тех пор, пока не получим частное, равное нулю;
2. Полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления;
3. Составить число в новой системе счисления, записывая его, начиная с последнего полученного остатка.

# Перевод из десятичной в любую систему

На первой «ступеньке» записываем исходное число. Под ступенькой – остаток от его деление на основание СС. На следующей ступеньке – результат целочисленного деления на основание СС. Продолжаем до тех пор, пока на ступеньке не окажется число, меньшее, чем основание СС. Оно и будет первой цифрой числа. Собираем разряды числа, начиная с конца «лестницы».



$$19_{10} = 10011_2$$

# Сложение в двоичной СС

В любой СС наибольшая цифра равна  $n-1$ , где  $n$  – основание этой СС. По этой причине легко оказаться в ситуации, когда необходимой цифры просто нет. Тогда стоит перейти на более старший разряд.

$$0 + 0 = 0_2$$

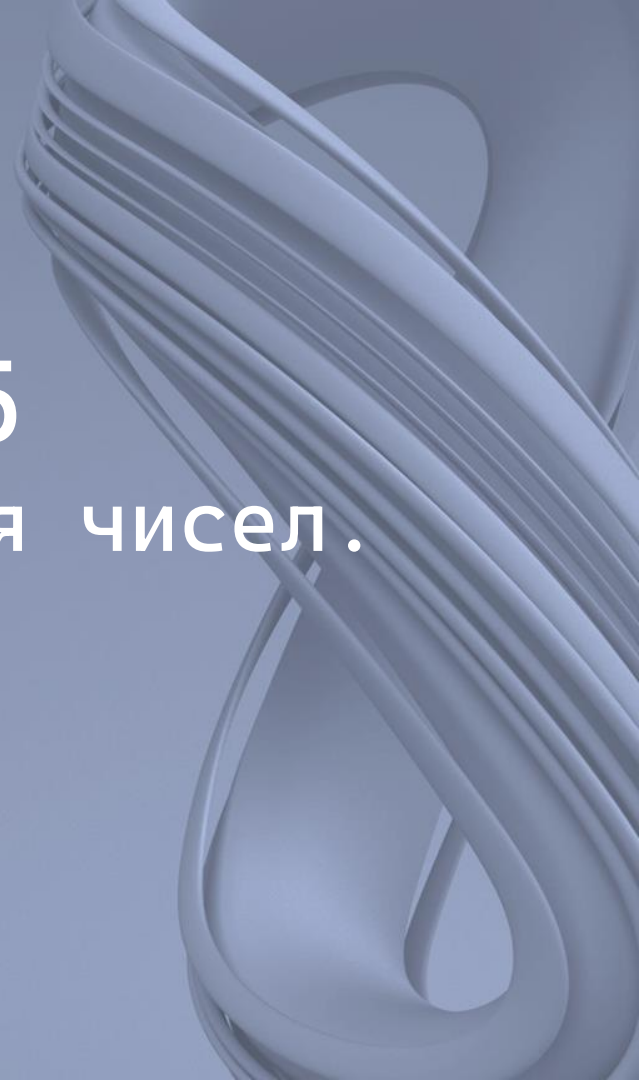
$$1 + 0 = 1_2$$

$$1 + 1 = 10_2$$



# Задание #5

## Анализ алгоритмов для чисел.



# Строки

Строки в Python – это неизменяемые объекты, которые состоят из последовательности символов. Строки заключаются в кавычки – одинарные (‘ ’) или двойные (“ ”).

Каждый символ строки имеет свой индекс, для первого символа он нулевой.

```
      0 1 2 3 4 5  
s = 'abcdef'
```

# Индексы строк

Для обращения к символу строки используются квадратные скобки, перед которыми указано имя переменной, хранящей строку. Внутри скобок указывается нужный индекс.

```
>>> s = 'abcdef'
```

```
>>> s[0]
```

```
'a'
```

```
>>> s[3]
```

```
'd'
```

```
      0 1 2 3 4 5  
s = 'abcdef'
```

# Индексы строк

Можно обратиться к конкретному символу с конца – тогда перед индексом необходимо поставить минус. При просмотре с конца строки, нумерация начинается с -1 (последний символ).

```
>>> s = 'abcdef'
```

```
>>> s[-1]
```

```
'f'
```

```
>>> s[-4]
```

```
'c'
```

```
      -6 -5 -4 -3 -2 -1  
s = 'abcdef'
```

# Срезы

```
>>> s = '1234567'  
>>> s[2:5]  
'345'
```

```
>>> s = '1234567'  
>>> s[2:]  
'34567'
```

```
>>> s = '1234567'  
>>> s[:5]  
'12345'
```

Срезы позволяют выделить часть строки (подстроку). Начало среза – индекс символа, включается в подстроку, а конец не включается. Также можно сделать срез с какого-то символа до конца строки или от начала строки до какого-то символа.

# Функция `int`

Функция `int()` в Python возвращает целочисленный объект из указанного ввода. Возвращаемый объект **всегда** будет в 10-ной системе счисления.

Если не указать вторым аргументом основание СС, то по умолчанию оно будет принято равным 10. Если же основание указать, то функция выполнит перевод в 10-ную СС.

```
>>> int('10') >>> int('10', 2) >>> int('17F', 16)
10           2           383
```

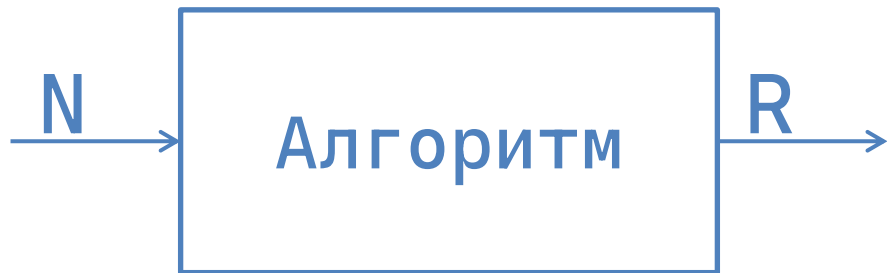
# Функция bin

Функция `bin()` в Python используется для преобразования целого числа в строку двоичного формата. Сформатированная строка имеет префикс «0b», который можно убрать, выполнив срез со второго символа и до конца строки.

```
>>> bin(10)
'0b1010'
```

```
>>> bin(10)[2:]
'1010'
```

В этом задании алгоритм получает на вход неизвестное десятичное число  $N$ . После обработки алгоритмом на выходе получаем результат – новое число  $R$ .





5

На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число  $R$  следующим образом.

1. Строится двоичная запись числа  $N$ .  
2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи числа  $N$ , и остаток от деления суммы на 2 дописывается в конец числа (справа).

*Например, запись 11100 преобразуется в запись 111001;*

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы её цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью результирующего числа  $R$ .

Укажите такое **наименьшее** число  $N$ , для которого результат работы данного алгоритма больше числа 77. В ответе это число запишите в десятичной системе счисления.

В двоичной системе счисления есть только две цифры: 0 и 1. Поэтому при сложении всех цифр числа сумма будет равна количеству единиц в двоичной записи числа.

# Метод count

Метод count возвращает количество вхождений подстроки в строку.

```
>>> s = '12125'  
>>> s.count('12')  
2
```

```
>>> s = '1011011'  
>>> s.count('1')  
5
```