

УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ  
НАПРАВЛЕНИЕ ПОДГОТОВКИ «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**  
**курса «Информатика»**

**Вариант 21**

Выполнил студент:

Берестовский Святослав Сергеевич  
группа: Р3111

Проверил:

·  
·

Санкт-Петербург, 2021 г.

# Содержание

<b>1. Перевод чисел между различными системами счисления</b>	<b>2</b>
1.1. Цель работы . . . . .	2
1.2. Задание . . . . .	2
1.3. Основная часть . . . . .	2
1.3.1. Исходный код . . . . .	2
1.3.2. Результат работы программы . . . . .	4
1.4. Выводы . . . . .	4
<b>Литература</b>	<b>5</b>

## Лабораторная работа 1

# Перевод чисел между различными системами счисления

### 1.1. Цель работы

Разобраться с принципом и основными техниками перевода чисел между разными системами счисления.

### 1.2. Задание

Перевести число  $A$ , заданное в системе счисления  $B$ , в систему счисления  $C$ . Для примеров с 5-го по 7-й выполнить операцию перевода по сокращенному правилу (для систем с основанием 2 в системы с основанием  $2^k$ ). Для примеров с 4-го по 6-й и с 8-го по 9-й найти ответ с точностью до 5 знака после запятой.

Таблица : Значения переменных

Задание	$A$	$B$	$C$
1	34106	10	15
2	16116	7	10
3	21104	5	15
4	51,96	10	2
5	41,6C	16	2
6	14,67	8	2
7	0,001101	2	16
8	0,001011	2	10
9	1B,08	16	10
10	42	10	Fib
11	147	-10	10
12	1000010101	Fib	10
13	1678	-10	10

### 1.3. Основная часть

#### 1.3.1. Исходный код

Исходный код скрипта доступен также на моем GitHub: <https://github.com/anim3boy/itmo-labs/blob/main/computer-science/script.py>.

## Листинг 1.1: Исходный код

```

1 from string import digits as numbers, ascii_uppercase as letters
2 from math import log
3 from itertools import product
4
5 def base_translate(source, a=10, b=10):
6     ALPHABET = numbers + letters
7
8     int_part_of_decimal, fract_part_of_decimal = 0, 0
9     int_part_of_source, fract_part_of_source = "", ""
10
11     if a != 10:
12         # a -> 10
13         int_part_of_source, fract_part_of_source = source.split(',')
14         else:
15             int_part_of_source = source
16
17     for i in enumerate(reversed(int_part_of_source)):
18         index, number = i
19         int_part_of_decimal += int(ALPHABET.index(number)) * (a ** index)
20
21     for i in enumerate(fract_part_of_source):
22         index, number = i
23         index = -(index + 1)
24         fract_part_of_decimal += int(ALPHABET.index(number)) * (a ** index)
25     decimal = int_part_of_decimal + fract_part_of_decimal
26
27     else:
28         decimal = float(source.replace(',', '.'))
29
30     # 10 -> b
31     if b == 10: return str(int(decimal * 10 ** 5) / 10 ** 5).replace('.', ',') if int(decimal) != decimal
32     else decimal
33
34     fract_part_of_decimal = decimal - int(decimal)
35     int_part_of_decimal = int(decimal)
36
37     int_part_of_result, fract_part_of_result = "", ""
38
39     for i in range(5):
40         fract_part_of_result += str(ALPHABET[int(fract_part_of_decimal*b)])
41         fract_part_of_decimal = fract_part_of_decimal*b - int(fract_part_of_decimal*b)
42
43     while int_part_of_decimal > 0:
44         int_part_of_decimal, m = divmod(int_part_of_decimal, b)
45         int_part_of_result += ALPHABET[m]
46
47     int_part_of_result = "0" if not int_part_of_result else int_part_of_result
48     while fract_part_of_result and fract_part_of_result[-1] == "0": fract_part_of_result =
49     fract_part_of_result[:-1].replace('0', '', 1)[:-1]
50     return '%s%s%s' % (int_part_of_result[:-1], ',' if fract_part_of_result else '', fract_part_of_result)
51
52 def translate(source, a=10, b=10):
53     if a == b: return source
54     if b == 'Fib':
55         return fib_encoding(source, int(a))
56     if a == 'Fib':
57         return fib_decoding(source, int(b))
58     a, b = int(a), int(b)
59     if a > 0 and b > 0 and (int(log(b) / log(a)) == log(b) / log(a) or int(log(a) / log(b)) == log(a) / log
60     (b)) \
61     and ',' not in source: #!!
62         # "easy way"
63         block_length = int(log(b, a))
64         padding = len(source) % block_length
65         if padding:
66             source = "0" * (block_length - padding) + source
67         blocks = [source[i : i + block_length] for i in range(0, len(source), block_length)]
68         return ''.join([base_translate(i, a, b) for i in blocks])
69     else:
70         return base_translate(source, a, b)
71
72 def fib_encoding(n, a):
73     if a != 10: n = int(translate(n, a, 10))
74     n = int(n)
75     ALPHABET = []
76     a, b = 0, 1
77     while a <= n:
78         ALPHABET.append(a)
79         a, b = b, a + b
80     ALPHABET = sorted(list(set(ALPHABET[1:])))
81
82     for i in product([0,1], repeat=len(ALPHABET)):
83         if sum(i[j] * ALPHABET[j] for j in range(len(ALPHABET))) == n and '11' not in ''.join(list(map(str,

```

```

82         i))) :
83             return ''.join(list(map(str, i)))[: -1]
84
85 def fib_decoding(n, b):
86     ALPHABET = []
87     a, b = 0, 1
88     for i in range(len(n) + 2):
89         ALPHABET.append(a)
90         a, b = b, a + b
91     ALPHABET = sorted(list(set(ALPHABET[1:])))
92
93     return sum(int(n[j]) * ALPHABET[: -1][j] for j in range(len(ALPHABET)))
94
95 with open('s.txt') as f:
96     s = list(map(lambda x: x[: -1].split(), f.readlines()))
97     print('{<10} {<3} {<3} {}'.format("A", "B", "C", "Ans"))
98     for i in s:
99         print('{<10} [{<3} -> {<3}] {}'.format(i[0], i[1], i[2], translate(i[0], i[1], i[2])))

```

Функции были реализованы с использованием алгоритмов из материалов лекции. Я проигнорировал ряд функций из стандартной библиотеки языка, например перевод чисел в десятичную СС с использованием `int()`

### 1.3.2. Результат работы программы

```

/mnt/c/wsl/itmo-labs/computer-science(main) » python3 script.py
A          B          C          Ans
34106      [10 -> 15 ]      A18B
16116      [7 -> 10 ]       4521
21104      [5 -> 15 ]       639
51,96      [10 -> 2 ]       110011,1111
41,6C      [16 -> 2 ]       1000001,01101
14,67      [8 -> 2 ]        1100,11011
0,001101   [2 -> 16 ]       0,34
0,001011   [2 -> 10 ]       0,17187
1B,08      [16 -> 10 ]      27,03125
42         [10 -> Fib]      10010000
147        [-10 -> 10 ]      67
1000010101 [Fib -> 10 ]      101
1678       [-10 -> 10 ]     -462

```

Рис. : Скриншот фрагмента сессии в ZSH

## 1.4. Выводы

В процессе выполнения данной лабораторной работы я систематизировал имеющиеся знания по переводу чисел между разными системами счисления, реализовал скрипт на питоне и поработал над читаемостью кода

# Литература

- [1] Алексеев Е.Г., Богатырев С.Д. Информатика. Мультимедийный электронный учебник.  
URL: <http://inf.ealekseev.ru/text/toc.html>
- [2] Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.: ил.