МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

«ИНФОРМАТИКА»

Перевод чисел между

различными системами счисления

Вариант №27

***Выполнил:***Студент группы P3107  
 Чусовлянов Максим Сергеевич

***Проверил:***Балакшин Павел Валерьевич

кандидат технических наук, доцент факультета ПИиКТ

**Содержание**

[Задание 3](#_gjdgxs)

[Основные этапы вычисления 4](#_30j0zll)

[1. 25307(10) = ?(9) 4](#_1fob9te)

[2. 10053(7) = ?(10) 4](#_3znysh7)

[3. 28D10(15) = ?(5) 4](#_2et92p0)

[4. 52,16(10) = ?(2) 4](#_tyjcwt)

[5. 3B,64(16) = ?(2) 5](#_3dy6vkm)

[6. 73,14(8) = ?(2) 5](#_1t3h5sf)

[7. 0,001001(2) = ?(16) 5](#_4d34og8)

[8. 0,011001(2) = ?(10) 5](#_2s8eyo1)

[9. 1F,1E(16) = ?(10) 5](#_17dp8vu)

[10. 75(10) = ?(Фиб) 5](#_3rdcrjn)

[11. 33{^2}00(7С) = ?(10) 5](#_26in1rg)

[12. 10100010(Фиб) = ?(10) 5](#_lnxbz9)

[13. 1000001,000001(Берг) = ?(10) 5](#_35nkun2)

[Листинг программы 6](#_1ksv4uv)

[Заключение 12](#_2jxsxqh)

[Список литературы 13](#_z337ya)

# **Задание**

Перевести число "А", заданное в системе счисления "В", в систему счисления "С".

# **Основные этапы вычисления**

## 1. 25307(10) = ?(9)

| **Число** | **Остаток** |
| --- | --- |
| 25307 | 8 |
| 2811 | 3 |
| 312 | 6 |
| 34 | 7 |
| 3 | 3 |

2530710 = 376389

## 2. 10053(7) = ?(10)

100537 = 3 \* 70 + 5 \* 71 + 0 \* 72 + 0 \* 73 + 1 \* 74 = 2401 + 0 + 0 + 35 + 3 = 243910

## 3. 28D10(15) = ?(5)

28D1015 = 0 \* 150 + 1 \* 151 + 13 \* 152 + 8 \* 153 + 2 \* 154 = 13119010

131190 / 5 = 26238 (ост: 0)

26238 / 5 = 5247 (ост: 3)

5247 / 5 = 1049 (ост: 2)

1049 / 5 = 209 (ост: 4)

209 / 5 = 41 (ост: 4)

41 / 5 = 8 (ост: 1)

8 / 5 = 1 (ост: 3)

1 / 5 = 0 (ост: 1)

28D1015 = 131442305

## 4. 52,16(10) = ?(2)

Перевод целой части:

5210 = 32 + 16 + 4 = 1101002

Перевод дробной части:

0,16 \* 2 = 0,32

0,32 \* 2 = 0,64

0,64 \* 2 = 1,28

0,28 \* 2 = 0,56

0,56 \* 2 = 1,12

53,5410 = 110100,001012

## 5. 3B,64(16) = ?(2)

316 = 00112

B16 = 10112

616 = 01102

416 = 01002

Целая часть:

111011

Дробная часть (с округление до 5 знаков после запятой): :0,011001 ≈ 0,0110

72.9816 = 111011,0112

## 6. 73,14(8) = ?(2)

78 = 1112

38 = 0112

18 = 0012

48 = 1002

Целая часть: 111011

Дробная часть (с округление до 5 знаков после запятой): 0,001100 ≈ 0,00110

73,148 = 111011,00112

## 7. 0,001001(2) = ?(16)

0.0010012 = 0.001001002 = 0.2416

## 8. 0,011001(2) = ?(10)

0.0110012 = 1 \* 2-2 + 1 \* 2-3 + 1 \* 2-6= 0.39062510 ≈ 0.3906310

## 9. 1F,1E(16) = ?(10)

1F,1E16 = 1 \* 161 + 15 \* 160 + 1 \* 16-1 + 14 \* 16-2 = 31.117187510 ≈ 31.1171910

## 10. 75(10) = ?(Фиб)

Ряд Фибоначчи: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, …

75 = 55 + 13 + 5 + 2

75(10) = 100101010(фиб)

## 11. 33{^2}00(7С) = ?(10)

33{^2}007С = 3 \* 74 + 3 \* 73 - 2 \* 72 = 8134

## 12. 10100010(Фиб) = ?(10)

Ряд Фибоначчи: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, …

10100010(Фиб) = 0\*1 + 1\*2 + 0\*3 + 0\*5 + 0\*8 + 1\*13 + 9\*21 + 1\*34 = 2 + 13 + 34 = 4910

## 13. 1000001,000001(Берг) = ?(10)

1000001.000001(Берг) = z6 + z0 + z-6 = 17.9442719… + 1 + 0.05572809… = 1910

# **Листинг программы**

from string import digits, ascii\_uppercase

from bisect import bisect

def gen\_fib(n):

x = [1, 2]

for i in range(2, n):

x.append(x[i-1]+x[i-2])

return x

def gen\_fact(n):

x = [1, 2]

for i in range(3, n):

x.append(i \* x[-1])

return x

ALP = digits + ascii\_uppercase

FIB = gen\_fib(100)

FACT = gen\_fact(20)

GOLDEN = (1 + 5 \*\* 0.5) / 2

PHI\_POS = [(pow(GOLDEN, i), i) for i in range(49, -1, -1)]

PHI\_NEG = [(pow(GOLDEN, i), i) for i in range(-1, -51, -1)]

def dec\_encode(n: str, base: int) -> str:

n = int(n)

res = ""

while n != 0:

remainder = n % base

n //= base

if (remainder < 0):

remainder += ((-1) \* base)

n += 1

res += ALP[remainder]

return res[::-1]

def dec\_decode(n: str, base: int) -> int:

res = 0

for i, elem in enumerate(n[::-1]):

res += base \*\* i \* ALP.index(elem)

return res

def fib\_encode(n: str) -> str:

n = int(n)

res = ['0' for \_ in range(100)]

while n > 0:

ind = bisect(FIB, n)

if n == FIB[ind]:

res[ind] == '1'

n = 0

else:

res[ind-1] = '1'

n -= FIB[ind-1]

res = res[::-1]

return "".join(res[res.index('1'):])

def fib\_decode(n: str) -> int:

res = 0

length = len(n)

for i in range(length):

if n[length-i-1] == "1":

res += FIB[i]

return res

def fact\_encode(n: str) -> str:

n = int(n)

divider = 2

res = []

while n > 0:

res.append(str(n % divider))

n //= divider

divider += 1

return ",".join(res[::-1])

def fact\_decode(n: str) -> int:

res = 0

tmp = list(map(int, n.split(".")))

length = len(tmp)

for i in range(length):

res += FACT[i] \* tmp[length-i-1]

return res

def berg\_encode(n: str) -> str:

n = int(n)

res1, res2 = ["0" for \_ in range(50)], ["0" for \_ in range(50)]

for i in range(50):

if (PHI\_POS[i][0] > n):

continue

res1[i] = "1"

n -= PHI\_POS[i][0]

for i in range(5):

if (n - PHI\_NEG[i][0] > -10\*\*-5):

res2[i] = "1"

n -= PHI\_NEG[i][0]

f\_part = "".join(res1[res1.index("1"):])

s\_part = "".join(res2[:50-res2[::-1].index("1")])

return f\_part + "." + s\_part

def berg\_decode(n: str) -> int:

f\_part, s\_part = n.split(".")

res = 0

for i in range(len(f\_part)):

if f\_part[i] == "1":

res += pow(GOLDEN, len(f\_part)-i-1)

for i in range(len(s\_part)):

if s\_part[i] == "1":

res += pow(GOLDEN, -1-i)

return round(res)

def sym\_encode(n: str, base: int) -> str:

n, base, res = int(n), int(base), []

while (n != 0):

rem = n % base

n //= base

if (rem > base//2):

rem = rem - base

n += 1

res.append(str(rem) if rem >= 0 else "{^" + str(-rem) + "}")

return "".join(res[::-1])

def sym\_decode(n: str, base: int) -> int:

res, ind, tmp = 0, 0, []

while ind < len(n):

if n[ind].isdigit():

tmp.append(int(n[ind]))

else:

tmp.append(-int(n[ind+2]))

ind += 3

ind += 1

for i, elem in enumerate(tmp[::-1]):

res += elem \* pow(base, i)

return res

def float\_encode(n: str, base: int) -> str:

f\_part, s\_part = n.split(".")

f\_part, s\_part = int(f\_part), int(s\_part) / (10 \*\* len(s\_part))

res = ""

while f\_part != 0:

remainder = f\_part % base

f\_part //= base

res += ALP[remainder]

if not res:

res += "0"

res = res[::-1] + "."

for \_ in range(5):

s\_part \*= base

pre\_calc = int(s\_part)

res += str(pre\_calc)

s\_part -= pre\_calc

return res

def float\_decode(n: str, base: int) -> int:

res = 0

f\_part, s\_part = n.split(".")

for i, elem in enumerate(f\_part[::-1]):

res += ALP.index(elem) \* base \*\* i

for i, elem in enumerate(s\_part):

res += ALP.index(elem) \* base \*\* (-(i+1))

return res

def decode(n, base) -> int:

if type(base) == int:

res = float\_decode(n, base) if ("," in n) or (

"." in n) else dec\_decode(n, base)

elif base.endswith("C"):

res = sym\_decode(n, int(base[:-1]))

elif base == "Fib":

res = fib\_decode(n)

elif base == "Berg":

res = berg\_decode(n)

elif base == "Fact":

res = fact\_decode(n)

else:

return ValueError

return res

def encode(n, base) -> int:

if type(base) == int:

res = float\_encode(n, base) if ("," in n) or (

"." in n) else dec\_encode(n, base)

elif base.endswith("C"):

res = sym\_encode(n, int(base[:-1]))

elif base == "Fib":

res = fib\_encode(n)

elif base == "Berg":

res = berg\_encode(n)

elif base == "Fact":

res = fact\_encode(n)

else:

return ValueError

return res

def convert(n, base1, base2):

return encode(str(decode(n, base1)), base2)

if \_\_name\_\_ == "\_\_main\_\_":

task = [

("25307", 10, 9),

("10053", 7, 10),

("28D10", 15, 5),

("52.16", 10, 2),

("3B.64", 16, 2),

("73.14", 8, 2),

("0.001001", 2, 16),

("0.011001", 2, 10),

("1F.1E", 16, 10),

("75", 10, "Fib"),

("33{^2}00", "7C", 10),

("10100010", "Fib", 10),

("1000001.000001", "Berg", 10),

]

for i, (n, base1, base2) in enumerate(task, 1):

res = convert(n, base1, base2)

print("{:>2}) {:<15} in {:>4} to {:<4} = {}".format(

i, n, base1, base2, res))

Результат выполнения программы: решение лабораторной работы на ЯП Python([рис.1](#_44sinio))

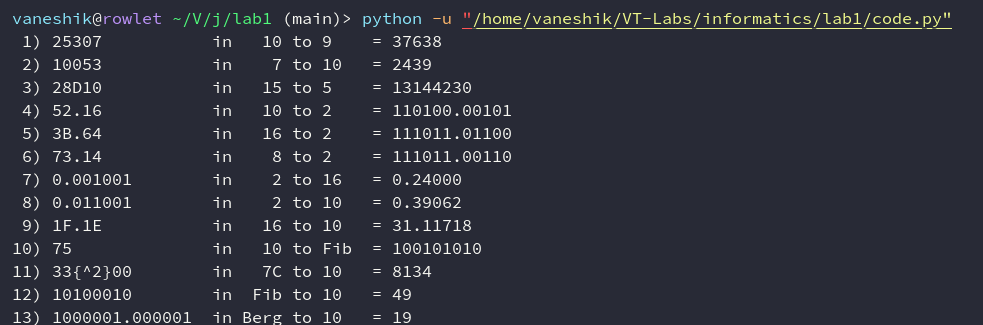


Рисунок 1 (Результат выполнения программы)

# **Заключение**

В ходе выполнения работы я закрепил свои знания по переводу чисел в классических системах счисления, узнал о фибоначчиевой и факториальной СС, СС Бергмана, а также научился записывать числа в них и переводить в другие системы счисления.

# **Список литературы**

1. Алексеев Е. Г., Богатырев С. Д. Информатика. Мультимедийный электронный учебник. Саранск: 2009.
2. С. Б. Гашков Системы счисления и их применение. 2 изд. М.: Издательство Московского центра непрерывного математического образования, 2012.