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**

«Перевод между различными системами счисления»

Вариант №36

Группа: P3132

Выполнил: Волков Г. А.

Проверил:

к.т.н. преподаватель Белозубов А.В.

г. Санкт-Петербург

2022г.

Оглавление

[Оглавление 2](#_Toc117664082)

[Текст задания 3](#_Toc117664083)

[Порядок выполнения работы 4](#_Toc117664084)

[Основные этапы вычисления 5](#_Toc117664085)

[Программа для перевода между системами счисления 8](#_Toc117664086)

[Работа программы на примере выполнения задания 13 13](#_Toc117664087)

[Заключение 14](#_Toc117664088)

[Список литературы 15](#_Toc117664089)

Текст задания

1. 83932(10) = X(15)
2. 87238(13) = X(10)
3. 4945C(13) = X(7)
4. 46.33(10) = X(2)
5. 68.76(16) = X(2)
6. 10.56(8) = X(2)
7. 0.011101(2) = X(16)
8. 0.010001(2) = X(10)
9. 8F.41(16) = X(10)
10. 676(10) = X(факт.)
11. 1001001(фиб.) = X(10)
12. 32{3}44(9C) = X(10)
13. 3088(10) = X(факт.)

Порядок выполнения работы

1. Перевести число "А", заданное в системе счисления "В", в систему счисления "С". Числа "А", "В" и "С" взять из представленных ниже таблиц. Вариант выбирается как сумма последних двух цифр в номере группы и номера в списке группы согласно ISU. Т.е. 13-му человеку из группы P3102 соответствует 15-й вариант (=02 + 13). Если полученный вариант больше 40, то необходимо вычесть из него 40. Т.е. 21-му человеку из группы P3121 соответствует 2-й вариант (=21 + 21 - 40).

2. Всего нужно решить 13 примеров. 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. В примере 11 группа символов {^1} означает -1 в симметричной системе счисления.

Основные этапы вычисления

1. 83932(10) = X(15)

83932 mod 15 = **7** 83932 div 15 = 5595

5595 mod 15 = **0** 5595 div 15 = 373

373 mod 15 = **13** = **D** 373 div 15 = 24

24 mod 15 = **9**

24 div 15 = **1**

**83932(10) = 19D07(15)**

1. 87238(13) = X(10)

8473223180(13) = 8 \* 134 + 7 \* 133 +2 \* 132 + 3 \* 131 + 8 \* 130 = 244252(10)

**87238(13) = 244252 (10)**

1. 4945C(13) = X(7)

44934251C0(13) = 4 \* 134 + 9 \* 133 +4 \* 132 + 5 \* 131 + 12 \* 130 = 134770(10)

134770 mod 7 = **6** 134770 div 7 = 19252

192552 mod 7 = **2** 192552 div 7 = 2750

2750 mod 7 = **6** 2750 div 7 = 392

392 mod 7 = **0** 392 div 7 = 56

56 mod 7 = **0** 56 div 7 = 8

8 mod 7 = **1**

8 div 7 = **1**

**4945С(13) = 1100626(7)**

1. 46.33(10) = X(2)

46.33(10) = 46(10) + 0.33(10)

46 mod 2 = **0** 46 div 2 = 23

23 mod 2 = **1** 23 div 2 = 11

11 mod 2 = **1**  11 div 2 = 5

5 mod 2 = **1** 5 div 2 = 2

2 mod 2 = **0**

2 div 2 = **1**

46(10) = 101110(2)

|  |  |  |
| --- | --- | --- |
| **0.** | 33 | \*2 |
| **0** | 66 | \*2 |
| **1** | 32 | \*2 |
| **0** | 64 | \*2 |
| **1** | 28 | \*2 |
| **0** | 56 | … |

0.33(10) = 0.01010(2)

**46.33(10)=101110.01010(2)**

Для 5-7 задания используем следующую таблицу перевода 2k систем счисления:

|  |  |  |  |
| --- | --- | --- | --- |
| X(8) | X(2) | X(16) | X(2) |
| 0 | 000 | 0 | 0000 |
| 1 | 001 | 1 | 0001 |
| 2 | 010 | 2 | 0010 |
| 3 | 011 | 3 | 0011 |
| 4 | 100 | 4 | 0100 |
| 5 | 101 | 5 | 0101 |
| 6 | 110 | 6 | 0110 |
| 7 | 111 | 7 | 0111 |
|  |  | 8 | 1000 |
|  |  | 9 | 1001 |
|  |  | A | 1010 |
|  |  | B | 1011 |
|  |  | C | 1100 |
|  |  | D | 1101 |
|  |  | E | 1110 |
|  |  | F | 1111 |

Таблица перевода между 2k системами

1. 68.76(16) = X(2)

**68.76(16) = 01101000.01110110(2)=1101000.0111011(2)**

1. 10.56(8) = X(2)

**10.56(8) = 001000.101110(2) = 1000.10111(2)**

1. 0.011101(2) = X(16)

|0000|.0111|0100|(2) = 0.74(16)

**0.011101(2) = 0.74(16)**

1. 0.010001(2) = X(10)

00.0-11-20-30-40-51-6(2) = 0 \* 20 + 0 \* 2-1 + 1 \* 2-2 + 0 \* 2-3 + 0 \* 2-4 + 0 \* 2-5 + 1 \* 2-6 = 0.25562…

**0.010001(2) = 0.25562 (10)**

1. 8F.41(16) = X(10)

81F0.4-11-2(16) = 8 \* 161 + 15 \* 160 + 4 \* 16-1 + 1 \* 16-2 = 143.25390…(10)

**8F.41(16) = 143.25390 (10)**

1. 676(10) = X(факт.)

676 mod 2 = **0** 676 div 2 = 338

338 mod 3 = **2** 338 div 3 = 112

112 mod 4 = **0** 112 div 4 = 28

28 mod 5 = **3**  28 div 5 = 5

5 mod 6 = **5**

5 div 6 = **0**

**676(10) = 53020(факт.)**

1. 1001001(фиб.) = X(10)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 21 | 13 | 8 | 5 | 3 | 2 | 1 |

1001001(фиб.)=21 + 5 + 1 = 27(10)

**1001001(фиб.) = 27(10)**

1. 32{3}44(9C) = X(10)

3423{3}24140(9C) = 3 \* 94 + 2 \* 93 - 3 \* 92 + 4 \* 91 + 4 \* 90 = 20938(10)

**32{3}44(9C) = 20938(10)**

1. 3088(10) = X(факт.)

3088 mod 2 = **0** 3088 div 2 = 1544

1544 mod 3 = **2** 1544 div 3 = 514

514 mod 4 = **2** 514 div 4 = 128

128 mod 5 = **3**  128 div 5 = 25

25 mod 6 = **1**

25 div 6 = **4**

**3088(10) = 413220(факт.)**

Программа для перевода между системами счисления

import math  
  
ph = (5\*\*0.5 + 1)/2  
alpha = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'

#Находим φ для перевода чисел из системы Бергмана и прописываем алфавит для перевода систем с основанием большем 10

def INTtoBASE(num, base1, base2):

#Если начальное основание не равно 10 переводим его в десятичную СС  
 if base1 != 10: pwr = 0  
 s = 0  
 num = num[::-1]  
 for i in num:  
 s += alpha.index(i) \* base1 \*\* pwr  
 pwr += 1  
 num = str(s)  
 num = int(num)  
 b = alpha[num % base2]  
 while num >= base2:  
 num = num // base2  
 b += alpha[num % base2]  
 return b[::-1]

#Функция для перевода целочисленного числа одной системы счисления в другую систему счисления

def FLOATtoBASE(num, base\_from, base\_to):

#Разделям число на целую и дробные части  
 n1, n2 = num.split('.')  
 s = 0

#Если начальное основание не равно 10 переводим его в десятичную СС  
 if base\_from != 10:  
 pwr = -1  
 s = 0  
 for i in n2:  
 s += alpha.index(i) \* base\_from \*\* pwr  
 pwr -= 1  
 n2 = str(s)[2:]  
 n1 = INTtoBASE(n1, base\_from, base\_to)  
 n2 = float('0.' + n2)  
 s = ''

#Переводим десятичную дробную часть в дробную часть системы base\_to до пяти знаков после запятой  
 for i in range(5):  
 n2 \*= base\_to  
 s += alpha[int(n2)]  
 n2 -= int(n2)  
 return str(n1) + '.' + s  
#Функция для перевода числа из десятичной системы счисления в факториальную систему счисления  
def DECtoFACT(num):  
 i = 2  
 a = ''  
 num = int(num)  
 while num // i > 0:  
 a += str(num % i)  
 num = num // i  
 i += 1  
 a += str(num + int(a[-1]) // (i-1))  
 return a[::-1]  
#Функция для перевода числа из факториальной системы счисления в десятичную

def FACTtoDEC(num):  
 num = str(num)[::-1]  
 s = 0  
 for i, e in enumerate(num):  
 s += e \* math.factorial(i + 1)  
 return s  
#Функция для перевода числа из симметричной системы счисления в десятичную  
def SIMtoDES(num, base):  
 pwr = 0  
 num = num[::-1]  
 s = 0  
 a = 1

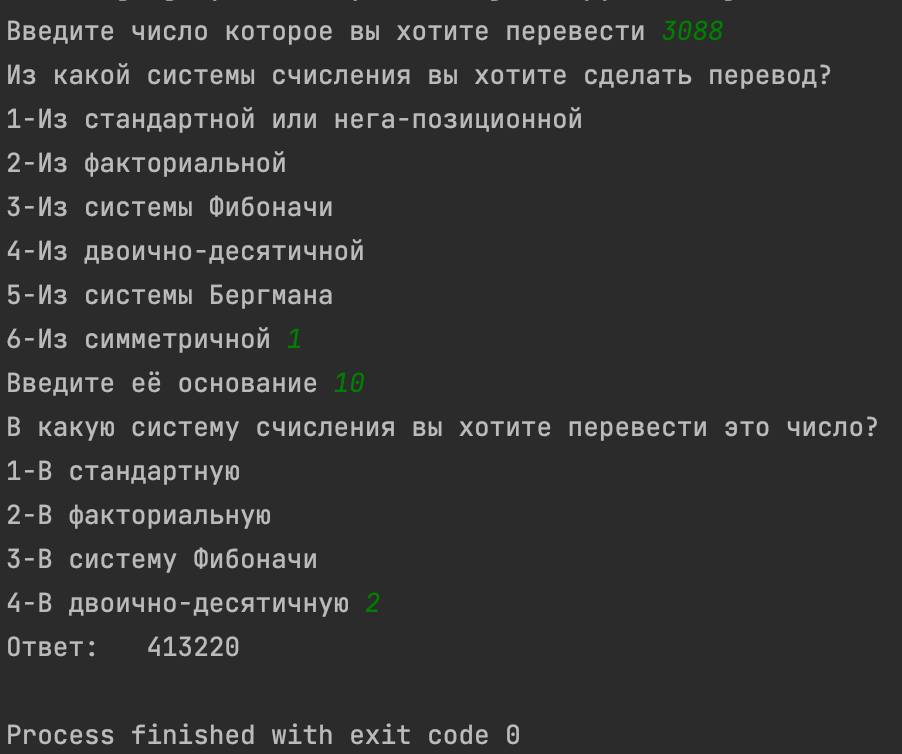
#Находим отрицательный элемент и переводим его из вида {x}  
 for i in num:  
 if i == '}':  
 a = -1  
 continue  
 elif i == '{':  
 a = 1  
 continue  
 s += a \* alpha.index(i) \* base \*\* pwr  
 pwr += 1  
 return str(s)  
  
def fib(n):  
 a, b = 0, 1  
 for i in range(n+1):  
 a, b = b, a + b  
 return b

#Функция для перевода числа из десятичной системы счисления в систему Фибоначчи  
  
def DEStoFib(num):  
 num = int(num)  
 fib = [1, 1]  
 while num >= fib[-1]:  
 fib.append(fib[-1]+fib[-2])  
 s ='1'  
 num -= fib[-2]  
 fib = fib[:-2]  
 for i in fib[::-1]:  
 if s[-1] == '1' or num < i:  
 s += '0'  
 else:  
 s += '1'  
 num -= i  
 return s  
#Функция для перевода числа из системы Фибоначчи в десятичную систему   
def FIBtoDES(num):  
 num = num[::-1]  
 s = 0  
 for i in range(0, len(num)):  
 s += int(num[i]) \* fib(i)  
 return str(s)  
#Функция для перевода числа из десятичной системы счисления в двоично-десятичную   
  
def DEStoDVDES(num):  
 s = ''  
 if '.' in num:  
 num, n2 = num.split('.')  
 for i in num:  
 s += str(INTtoBASE(i, 10, 2))  
 s += '.'  
 for i in n2:  
 s += str(INTtoBASE(i, 10, 2))  
 return str(float(s))  
 else:  
 for i in num:  
 s += str(INTtoBASE(i, 10, 2))  
 return str(int(s))  
#Функция для перевода числа из двоично-десятичной системы счисления в десятичную  
def DVDEStoDES(num, lv):  
 s = ''  
 n2 = ''  
 if '.' in num:  
 num, n2 = num.split('.')  
 if len(num) % lv != 0:  
 num = '0' \* (lv - (len(num) % lv)) + num  
 chunks = [num[i:i + lv] for i in range(0, len(num), lv)]  
 if len(n2) % lv != 0:  
 n2 = n2 + '0' \* (lv - (len(n2) % lv))  
 ch2 = [n2[i:i + lv] for i in range(0, len(n2), lv)]  
 for i in chunks:  
 s += INTtoBASE(i, 2, 10)  
 s += '.'  
 for i in ch2:  
 s += INTtoBASE(i, 2, 10)  
 return str(float(s))  
#Функция для перевода числа из системы Бергмана в десятичную  
def BERGtoDES(num):  
 if '.' in num:  
 num, n2 = num.split('.')  
 pwr = len(num)  
 num = num + n2  
 else:  
 pwr = len(num)  
 s = 0  
 for i in num:  
 pwr -= 1  
 s += int(i) \* ph \*\* pwr  
 return str(s)

#Ввод значений, выбор систем счислений для перевода. Вывод ответа  
def proverkanum(num):  
 for i in num:  
 if (i not in alpha and i not in prov) or (num[0] == '.') or (num.count('.') > 1):  
 print(  
 'Число может состоять из арабских цифр, букв латинского алфавита, а ткаже специальных символов ".", "{" и "}".\n'  
 'Причём символы "{" и "}" могут использоватьтся только при переводе из симметричной системы счисления и использоваться\n'  
 'в формате "\*{"цифра"}\*. Специальный символ "." не может стоять в числе первым или быть в числе в количестве больше 1')  
 return False  
 break  
 else:  
 return True  
def proverkabase(base):  
 for i in base:  
 if i not in '0123456789':  
 print('Основание должно быть числом от 2 до 36')  
 return False  
 break  
 else:  
 return True  
def proverkasim(num, base):  
 if base % 2 == 0:  
 print('Основание симметричной системы должно быть нечётным')  
 return False  
 elif '{}' in num or '}{' in num:  
 print('Фигурные скобки должны использоваться в формате \*{"цифра"}\*')  
 c = 0  
 for i in num:  
 if i == '{':  
 c += 1  
 elif c != 0 and i != '}':  
 print('Фигурные скобки должны использоваться в формате \*{"цифра/буква латинского алфавита"}\*')  
 return False  
 break  
 else:  
 if alpha.index(i) > base:  
 print('Число не может иметь данное основание')  
 return False  
 else:  
 return True  
def proverkalegal(num, base):  
 for i in num:  
 if i == '.':  
 continue  
 elif i in '{}':  
 print('Фигурные скобки используются только в симметричной системе счисления')  
 return False  
 break  
 elif alpha.index(i) >= int(base):  
 print('Число не может иметь такое основание или быть в данной системе счисления ')  
 return False  
 break  
 return True  
def proverkafact(num):  
 c = 1  
 for i in reversed(num):  
 if alpha.index(i) > c:  
 print('Число записано в неправильном формате ')  
 return False  
 break  
 else:  
 c += 1  
 return True

a = input('Из какой системы счисления вы хотите сделать перевод?\n'  
 '1-Из стандартной \n'  
 '2-Из факториальной\n'  
 '3-Из системы Фибоначи\n'  
 '4-Из двоично-десятичной\n'  
 '5-Из системы Бергмана \n'  
 '6-Из симметричной ')  
  
if a == '1':  
 base = (input('Введите основание системы '))  
 if not proverkabase(base):  
 quit()  
 elif not proverkalegal(num, base):  
 quit()  
 else:  
 base = int(base)  
elif a == '2':  
 if '.' in num:  
 print('В данной системе не может быть дробного числа')  
 quit()  
 elif not proverkalegal(num, len(alpha)-1):  
 quit()  
 elif not proverkafact(num):  
 quit()  
elif a in '345':  
 base = 2  
 if not proverkalegal(num, base):  
 quit()  
 elif a == '3' and '.' in num:  
 print('В данной системе не может быть дробного числа')  
 quit()  
elif a == '6':  
 base = (input('Введите основание симметричной системы '))  
 if not proverkabase(base):  
 quit()  
 base = int(base)  
else:  
 print('Такого "a" нет, попробуйте ещё ')  
 quit()  
  
  
if a == '1':  
 if '.' in num:  
 ans = FLOATtoBASE(num, base, 10)  
 else:  
 ans = INTtoBASE(num, base, 10)  
elif a =='2':  
 ans = FACTtoDEC(num)  
elif a =='3':  
 ans = FIBtoDES(num)  
 if DEStoFib(ans) != num:  
 print('Запись числа Фиббоначи неправильная. Правильная запись: ', DEStoFib(FIBtoDES(num)))  
  
elif a == '4':  
 ans = DVDEStoDES(num, int(input('Введите по сколько символов брать ')))  
elif a == '5':  
 if '11' in num:  
 num, num1 = num.split('.')  
 numberg = num  
 while '11' in numberg:  
 numberg = numberg.replace('11', '100')  
 num = num + '.' + num1  
 numberg = numberg + '.' + num1  
 print('Неправильная запись числа Бергмана, правильная запись: ', numberg)  
 ans = BERGtoDES(num)  
elif a == '6':  
 if not proverkasim(num, base):  
 quit()  
 ans = SIMtoDES(num, base)  
  
b = input('В какую систему счисления вы хотите перевести это число?\n'  
 '1-В стандартную\n'  
 '2-В факториальную\n'  
 '3-В систему Фибоначи\n'  
 '4-В двоично-десятичную ')  
  
if b == '1':  
 baseto = input('Введите её основание ')  
 if not proverkabase(baseto):  
 quit()  
 if '.' in str(ans):  
 print('Ответ: ', znak, FLOATtoBASE(ans, 10, int(baseto)))  
 else:  
 print('Ответ: ', znak, INTtoBASE(ans, 10, int(baseto)))  
elif b =='2':  
 print('Ответ: ', znak, DECtoFACT(ans))  
elif b =='3':  
 print('Ответ: ', znak, DEStoFib(ans))  
elif b == '4':  
 print('Ответ: ', znak, DEStoDVDES(ans))  
else:  
 print('Такого "b" нет, попробуйте ещё ')  
 quit()

Работа программы на примере выполнения задания 13



Заключение

Я освоил навыки между различными системами счисления. Научился переводу дробных чисел, переводу в факториальную и симметричную системы, системы Бергмана и Цекендорфа. Во время выполнения лабораторной работы я переводил числа в различные системы счисления с помощью представления чисел в виде полинома для перевода в десятичную, с помощью деления с остатком переводил числа из десятичной системы, сокращенным способом переводил числа в системах 2k. Также написал программу, с помощью которой можно переводить числа между различными системами счисления.

Список литературы

Орлов С. А., Цилькер Б. Я. Организация ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2011. – 688 с.: ил.

Алексеев Е.Г., Богатырев С.Д. Информатика. Мультимедийный электронный учебник. – Режим доступа: http://inf.e- alekseev.ru/text/toc.html.