

Рабочая тетрадь № 2

Система счисления – это символический метод записи чисел.

Непозиционные системы – ранние системы счисления. В этих системах каждая цифра имеет значение, не зависящее от положения.

Позиционные системы – значение каждой цифры зависит от ее положения (разряда) в записи числа.

1. Теоретический материал

Чтобы любое число в k -ичной системе счисления перевести в десятичную систему счисления нужно воспользоваться формулой [1, 3]:

$$X_{10} = a_0k^0 + a_1k^1 + \dots + a_Nk^N,$$

если $X_k = a_N \dots a_2 a_1 a_0$.

2. Пример

Задача:

Дано: $X_{10} = E8A_{16}$. Найти X_{10} .

Решение:

$$X_{10} = 10 + 8 * 16 + 14 * 16^2 = 3722$$

Ответ:

$$X_{10} = 3722$$

3. Задания

1. **Задача:**

Дано: $X_{10} = 101010_2$. Найти X_{10} .

Решение:

$$x(10) = 0*1 + 1*2 + 0*4 + 1*8 + 0*16 + 1*32 = 2+8+32=42$$

Ответ:

$$x(10) = 42$$

2. **Задача:**

Дано: $X_{10} = 563_8$. Найти X_{10} .

	Решение:
	$x(10) = 3 \cdot 1 + 6 \cdot 8 + 5 \cdot 64 = 3 + 48 + 320 = 371$
	Ответ:
	371
3.	Задача:
	Дано: $X_{10} = A1286_{12}$. Найти X_{10} .
	Решение:
	$x(10) = 6 \cdot 1 + 8 \cdot 12 + 2 \cdot 144 + 1 \cdot 1728 = 1728 + 288 + 96 + 6 = 2118$ $x(10) = 6 \cdot 1 + 8 \cdot 12 + 2 \cdot 144 + 1 \cdot 1728 + A \cdot 20736 = 207360 + 1728 + 288 + 96 + 6 = 209478$
	Ответ:
	$x(10) = 2118$ $x(10) = 209478$
4.	Задача:
	Сколько единиц в двоичной записи числа 127?
	Решение:
	<input type="text"/>
	Ответ:
	7

1. Теоретический материал

Чтобы число X из десятичной системы перевести в k -ичную, нужно:

1. Разделить X на k : пусть X_1 – это целая часть отношения, а a_0 – остаток от деления.

2. Если X_1 не равно нулю, то делим X_1 на k , обозначаем через X_2 целую часть, через a_1 – остаток.

3. Деление происходит до тех пор, пока частное не станет меньше основания системы счисления.

В результате

$$X = a_N a_{(N-1)} \dots a_1 a_0 ,$$

есть представление в k -ичной системе счисления.

2. Пример

Задача:

Дано: $48_{10} = X_3$. Найти X .

Решение:

48_{10} делим на 3, частное = 16, остаток $a_0 =$

частное = 16_{10} делим на 3, частное = 5, остаток $a_1 = 1$

частное = 5_{10} делим на 3, частное = 1, остаток $a_2 = 2$

частное = 1_{10} делим на 3, частное = 0, остаток $a_3 = 1$

Частное не больше нуля, деление закончено. Для представления числа в заданной системе счисления остатки от деления записываются в обратном порядке:

$$48_{10} = (a_3 a_2 a_1 a_0)_3 = 1210_3.$$

Ответ:

$$48_{10} = 1210_3.$$

3. Задания

1. Задача:

Дано: $367_{10} = X_7$. Найти X .

Решение:

$tmp = 367 // 7 = 52, a_1 = 367 \% 7 = 3tmp$

$$(a_4 a_3 a_2 a_1)_7 = 1033(7)$$

Ответ:

$$x = 1033(7)$$

2. Задача:

Дано: $1143_{10} = X_{12}$. Найти X .

Решение:

$tmp = 1143 // 12 = 95, a_1 = 1143 \% 12 = 3tmp = 9$

$$(a_3 a_2 a_1)_{12} = 7B3(12)$$

Ответ:

$$x = 7B3(12)$$

3. **Задача:**

Дано: $127_8 = X_9$. Найти X .

Решение:

$127(8) = 7 \cdot 16 + 1 \cdot 8 + 7 = 112 + 8 + 7 = 127$

$(a3a2a1)_9 = 103(9)$

Ответ:

$x = 103(9)$

4. **Задача:**

Дано: $AB4_{13} = X_6$. Найти X .

$AB4(13) = 4 \cdot 169 + B \cdot 13 + A \cdot 1 = 676 + 13B + A$

$(a5a4a3a2a1)_6 = 12301(6)$

$tmp = 1837 // 6 = 306, a1 = 1837 \% 6 = 1tmp$

Ответ:

12301

1. Теоретический материал

Перевод чисел между системами счисления, основания которых равны значениям степеней числа 2, можно произвести по более простым алгоритмам.

Нетрудно заметить, что информационный вес восьмеричной цифры в три раза больше двоичного. Поэтому каждой восьмеричной цифре можно поставить в соответствие группу из трех двоичных разрядов (триаду). Информационный вес шестнадцатеричной цифры в четыре раза больше двоичного. Значит, каждой цифре шестнадцатеричной системы счисления можно поставить в соответствие группу из четырех двоичных разрядов (тетраду). Ниже в таблице приведено записи чисел в системах счисления с основанием, равным степени двойки

десятичная	двоичная	восьмеричная	шестнадцатеричная
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6

4

7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	101	15	D
14	1110	16	E
15	1111	17	F

Алгоритм перевода двоичного числа в восьмеричную систему счисления:

1. Разбить двоичное число на триады, справа налево.
2. Если в правой группе меньше трех цифр, то добавить ведущие нули.
3. Каждую триаду перевести в восьмеричную систему счисления.
4. Для получения итогового числа в восьмеричной системе счисления произвести запись цифр в соответствующих разрядах.


Алгоритм перевода восьмеричного числа в двоичную систему счисления:

1. Разбить двоичное число на триады, справа налево.
2. Поставить в соответствие каждой восьмеричной цифре двоичную триаду.
3. Соединить триады и записать двоичное число.
4. Удалить (если существуют) незначащие нули.

Для перевода из двоичной системы счисления в шестнадцатеричную из шестнадцатеричной в двоичную алгоритм аналогичен, за тем исключением, что вместо трех разрядов необходимо использовать четыре.

2. Пример

Задача:

 Перевести двоичное число 1011101110_2 в восьмеричную систему счисления.

Решение:

 Для решения задачи воспользуемся выше приведенным алгоритмом:

1. **1.011.101.110**
2. **001.011.101.110**
3. **1 3 5 6**
4. **$1011101110_2 = 1356_8$**

Ответ:	
<input type="checkbox"/>	1356
Задача:	
<input type="checkbox"/>	Перевести шестнадцатеричное число $3AC_{16}$ в двоичную систему счисления.
Решение:	
<input type="checkbox"/>	Используем алгоритм, приведенный выше: 1. 0011.1010.1100 2. 001110101100 3. 1110101100 Таким образом, $3AC_{16} = 1110101100_2$
Ответ:	
<input type="checkbox"/>	1110101100



3. Задания	
1.	Задача:
<input type="checkbox"/>	Дано: $1010011101_2 = X_{16}$. Найти X.
Решение:	
<input type="checkbox"/>	1. 10.1001.1101 4. $1010011101(2) = 29D(16)$ 2. 0010.1001.1101
Ответ:	
<input type="checkbox"/>	0x29D
2.	Задача:
<input type="checkbox"/>	Дано: $1147453_8 = X_2$. Найти X.
Решение:	
<input type="checkbox"/>	1) 1 1 4 7 4 5 3 2) 001.001.100.111.100.101.011 4) $1147453(8) = 1001100111100101011(2)$
Ответ:	
<input type="checkbox"/>	0b1001100111100101011
3.	Задача:
<input type="checkbox"/>	Дано: $127_8 = X_{16}$. Найти X.
Решение:	
<input type="checkbox"/>	127 = 001.010.111 = 1010111 = 0101.0111 = 0x57
Ответ:	
<input type="checkbox"/>	0x57

4.	<p>Задача:</p> <p>Сколько значащих нулей в двоичной записи шестнадцатеричного числа $C3E1_{16}$?</p> <p>Решение:</p> <p></p> <p>Ответ:</p> <p>8</p>
5.	<p>Задача*:</p> <p>Дано: $AF38_{32} = X_{16} = X_8 = X_2$. Найти X.</p> <p>Решение:</p> <p>$AF38(32) = 01010.01111.00011.01000 = 1010011110001101000(2)1010011110001101000(8)001236150(16)$</p> <p>Ответ:</p> <p>$X(16) = 0x53C68$; $X(8) = 001436144$; $X(2) = 0b1010011110001101000$</p>
6.	<p>Задача*:</p> <p>Некоторое число X из десятичной системы счисления перевели в системы счисления с основаниями 16, 8, 4, 2. Часть символов при записи утеряна. Позиции утерянных символов обозначены знаком *</p> $X = E*_{16} = *5*_8 = ***1_4 = *****1**_2$ <p>Определите число X.</p> <p>Решение:</p> <p>1) 1110.**** 2) 11.101.*** 3) 11.10.1*.01 4) 1.1.1.0.1.1.0.1 = $0b11101101 = 237$</p> <p>Ответ:</p> <p>$X(10) = 237$</p>



Тест 2

1.	<p>Задание:</p> <p>Переведите число 37 из десятичной системы счисления в двоичную:</p> <p>Ответ:</p> <p>A) 100101; B) 10101; C) 10011; D) 101101.</p>
2.	<p>Задание:</p> <p>Переведите число 11010_2 из двоичной системы счисления в десятичную систему счисления.</p> <p>Ответ:</p> <p>A) 18; B) 24; C) 26; D) 14.</p>

3.	Задание:
	<input type="checkbox"/> Дано: $a = D7_{16}$, $b = 331_8$. Какое из чисел c , записанных в двоичной системе, отвечают условию $a < c < b$?
	Ответ:
4.	<input type="checkbox"/> A) 11011001 B) 1101110 C) 11010111 D) 11011000
	Задание:
	<input type="checkbox"/> Для чисел, заданных в различных системах счисления: $X = 1112_3$, $Y = 140_5$, $Z = 222_4$ – справедливо соотношение:
5.	Ответ:
	<input type="checkbox"/> A) $X < Y < Z$ B) $X < Z < Y$ C) $Y < X < Z$ D) $Y < Z < X$
	Задание:
	<input type="checkbox"/> В системе счисления с некоторым основанием десятичное число 144 записывается в виде 264. Укажите это основание.
6.	Ответ:
	<input type="checkbox"/> A) 5 B) 6 C) 7 D) 8
	Задание:
	<input type="checkbox"/> Укажите основание системы счисления, которой не может быть записано число 1302
7.	Ответ:
	<input type="checkbox"/> A) 3 B) 10 C) 7 D) 4
	Задание:
	<input type="checkbox"/> Значение выражения $10_{16} + 10_8 * 10_2$ в двоичной системе счисления равно
8.	Ответ:
	<input type="checkbox"/> 1) 1010 B) 11010 C) 100000 D) 110000
	Задание:
	<input type="checkbox"/> Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 40 оканчивается на 4.
2.	Ответ:
	<input type="checkbox"/>
	Задание:
	<input type="checkbox"/> В системе счисления с некоторым основанием десятичное число 52 записывается в виде 202_n . Найдите n .
	Ответ:
	<input type="checkbox"/>

3.	Задание:
	Решите уравнение: $12_6 + x = 324_5$ Ответ запишите в десятичной системе счисления.
	Ответ:
	

Реализация задач на языке программирования Python

1. Теоретический материал	
<p>Для перевода числа из одной системы счисления в другую в Python существует несколько функций:</p> <ul style="list-style-type: none"> • int([object], [основание системы счисления]) - преобразование к целому числу в десятичной системе счисления. По умолчанию система счисления десятичная, но можно задать любое основание от 2 до 36 включительно. • bin(x) - преобразование целого числа в двоичную строку. • hex(x) - преобразование целого числа в шестнадцатеричную строку. • oct(x) - преобразование целого числа в восьмеричную строку. 	
2. Пример	
Задача:	
	Ввести число в десятичной системе счисления. Вывести двоичную, восьмеричную и шестнадцатеричную запись введенного числа
Решение (код программы):	
	<pre>print('Введите число в десятичной системе счисления') a = int(input()) print('Двоичная: ', bin(a)) print('Восьмеричная: ', oct(a)) print('Шестнадцатеричная: ', hex(a))</pre>

3. Задания

Задача:

На вход программа получает две величины: n , A , где n – натуральное число от 2 до 36, основание системы счисления, A – число, записанное в системе счисления с основанием n , $A < 2^{31}$.

Необходимо вывести значение A в системе счисления с основанием десять. В задаче подразумевается корректный ввод (т.е. в числе A отсутствуют цифры большие или равные n).

Решение (код программы):

```
n, A = input("base, num: ").split(" "); alph = "0123456789abcdefghijklmnopqrstuvwxyz"res
```

```
n, A = input("base, num: ").split(" ");  
print(int(A,int(n)))
```

Все ранее рассматриваемые программы имели линейную структуру: все инструкции выполнялись последовательно одна за одной, каждая записанная инструкция обязательно выполняется.

Оператор ветвления *if* позволяет выполнить определенный набор инструкций в зависимости от некоторого условия.

1. Теоретический материал

Синтаксис оператора *if* в *Python* выглядит следующим образом:

if выражение:

инструкция_1

инструкция_2

...

инструкция_n

После оператора *if* записывается выражение. Если это выражение истинно, то выполняются инструкции, определяемые данным оператором.

Стоит отметить особенность языка *Python*. Он не содержит операторных скобок (`begin..end` в *pascal* или `{..}` в *Си*), вместо этого **блоки выделяются отступами**: четырем пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием.

Бывают случаи, когда необходимо предусмотреть альтернативный вариант выполнения программы. Т.е. при истинном условии нужно выполнить один набор инструкций, при ложном – другой. Для этого используется

конструкция *if – else*. Для реализации выбора из нескольких альтернатив можно использовать конструкцию *if – elif – else*.

```
if выражение_1:
    инструкции_(блок_1)
elif выражение_2:
    инструкции_(блок_2)
elif выражение_3:
    инструкции_(блок_3)
else:
    инструкции_(блок_4)
```

2. Пример

Задача:

Напечатать модуль введенного числа

Решение (код программы):


```
x = int(input('Введите x')) # преобразуем строку в целое число
if x < 0: # если введенное число меньше нуля
    x = -x
print(x)
```







Задача:

Ввести два числа и определить четверть координатной плоскости

Решение (код программы):

```
x = int(input())
y = int(input())
if x > 0 and y > 0:
    print("Первая четверть")
elif x > 0 and y < 0:
    print("Четвертая четверть")
elif y > 0:
    print("Вторая четверть")
else:
```

	<code>print("Третья четверть")</code>
---	---------------------------------------

3. Задания	
1.	<p>Задача:</p> <p> Дано двузначное число. Определить входит ли в него цифра 3. (// - операция получения целой части от деления, % - операция взятия остатка от целочисленного деления).</p> <p>Решение (код программы):</p> <div>  <pre>print("has '3': {}".format("3" in input("number: "))) num = int(input("number: ")) while(num > 0):</pre> </div>
2.	<p>Задача:</p> <p> Дано двузначное число. Определить какая из его цифр больше.</p> <p>Решение (код программы):</p> <div>  <pre>print("max digit: {}".format(max([int(i) for i in input("number: ")]))) num = int(input("number: ")); max_d = 0 if((num // 10) > (num % 10)): max_d = num // 10 else: max_d = num % 10 print("max digit: {}".format(max_d))</pre> </div>
3.	<p>Задача:</p> <p> Найти корни квадратного уравнения и вывести их на экран, если они есть. Если корней нет, то вывести сообщение об этом. Конкретное квадратное уравнение определяется коэффициентами a, b, c, которые вводит пользователь.</p> <p>Решение (код программы):</p> <div>  <pre>a,b,c = [int(i) for i in input("a,b,c: ").split(" ")] D = (b**2) - 4*a*c if(D < 0): print("no roots"); exit(1) roots = set([str((-b+(D**0.5))/(2*a)), str((-b-(D**0.5))/(2*a))]) print("roots: {}".format(" ".join(roots)))</pre> </div>

Цикл –конструкция языка программирования, предназначенная для организации многократного исполнения набора команд (инструкций).

При этом такая последовательность инструкций называется телом цикла.

Единичное выполнение тела цикла называется итерацией.

Выражение, определяющее, будет в очередной раз выполняться итерация или цикл завершится, называется условием выхода или условием окончания цикла.

1. Теоретический материал

Оператор цикла **while** выполняет указанный набор инструкций до тех пор, пока условие цикла истинно. Истинность условия определяется как и в случае оператора **if**. Синтаксис оператора **while** выглядит следующим образом.

while выражение:

```
инструкция_1  
инструкция_2  
...  
инструкция_n
```

Оператор **for** выполняет указанный набор инструкций заданное количество раз, которое определяется количеством элементов в наборе. Например:

```
for i in [1,2,3,4,5]:  
    a = i * i  
print(a)
```

В результате на экран будут выведены квадраты чисел от одного до пяти.

Переменная цикла **i** последовательно принимает все значения заданного списка, при этом каждый раз выполняется блок операторов, выделенный отступами. При создании цикла удобно пользоваться функцией **range(a,b)**, которая создает последовательность чисел от **a** до **b-1**. Пример:

```
for i in range(1, 6):  
    print("Hello")
```

В результате «Hello» будет выведено пять раз.

При выполнении цикла часто возникает необходимость досрочного прекращения выполнения цикла и пропустить какую-либо итерацию. Для этого используются конструкции **break** и **continue**. Оператор **continue** начинает следующий проход цикла, минуя оставшееся тело цикла (**for** или **while**), оператор **break** досрочно прерывает цикл.

2. Пример

Задача:

Выведите все точные квадраты натуральных чисел, не превосходящие данного числа N .

Решение (код программы):

```
n=int(input())
i=1
while i**2<n:
    print (i**2)
    i+=1
```

Задача:

Вывести квадраты чисел от нуля до девяти

Решение (код программы):

```
for i in range(10):
    a = i * i
    print(a)
```

Задача:

Напишите программу, которая выводит чётные числа из заданного списка и останавливается, если встречается число 5.

Решение (код программы):

```
n = [1, 2, 3, 7, 6, 4, 5, 8] #примерсписка
for x in n:
    if x == 237:
        break
    elif x % 2 == 0:
        print(x)
```

Задача:

Ввести строку. Вывести на экран все символы строки кроме пробелов

Решение (код программы):

```
s = input()
for i in s:
```

```

if(i == ' '):
    continue
print(i, end = '') # end = '' непереводит на новую строку

```

3. Задания

1. Задача:

Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

Решение (код программы):

```
num = int(input("number: "))
for i in range(2, num+1):
    if(num % i == 0):
        print(i)
        break
```

2. Задача:

Посчитать сумму числового ряда от 1 до N включительно (т.е. $0+1+2+3+\dots+N$). Решите задачу с помощью и без помощи оператора цикла. Число N вводится с клавиатуры ($N < 1000$).

Решение (код программы):

```

N = int(input("last element: "))
print("sum: {0}".format(sum([i for i in range(N+1)])))

N = int(input("last element: "))
print("sum: {0}".format(((0+N)/2)*(N+1)))

```

3. Задача:

Простыми являются натуральные числа больше 1, которые делятся нацело только на 1 и самих себя. На вход программе подается число. Необходимо проверить является ли оно простым.

Решение (код программы):

```
num = int(input())
if(n <= 1):
    print(False)
else:
    for i in range(2, num):
        if(num % i == 0):
            print(False)
            break
    else:
        print(True)
```

4. Задача:

Вводится десятичное число A ($A < 2^{31}$) и число n ($2 \leq n \leq 9$). Необходимо перевести введенное число A в систему счисления с основанием n . При этом использовать встроенные конструкции языка Python не разрешается.

Решение (код программы):

```
def to_n(num, base):
    alph = "0123456789abcdefghijklmnopqrstuvwxyz"
    res = ""
    while num > 0:
        res = alph[num % base] + res
        num = num // base
    return res
```