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]:  X10 = a0k0 + a1k1 + … + aNkN,  если xk = aN…a2a1a0. | |
| **2. Пример** | |
| ***Задача:*** | |
|  | Дано: X10 = E8A16. Найти X10. |
| ***Решение:*** | |
|  | X10 = 10 + 8 \* 16 + 14 \* 162 = 3722 |
| ***Ответ:*** | |
|  | X10 = 3722 |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Дано: X10 = 1010102. Найти X10. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 42 |
| 2. | ***Задача:*** | |
|  | Дано: X10 = 5638. Найти X10. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 371 |
| 3. | ***Задача:*** | |
|  | Дано: X10 = A128612. Найти X10. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 209478 |
| 4. | ***Задача:*** | |
|  | Сколько единиц в двоичной записи числа 127? |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 7 |

|  |
| --- |
| **1. Теоретический материал** |
| Чтобы число X из десятичной системы перевести в k-ичную, нужно:  1. Разделить X на k: пусть X1 – это целая часть отношения, а a0 – остаток от деления.  2. Если X1 не равно нулю, то делим X1 на k, обозначаем через X2 целую часть, через a1 – остаток.  3. Деление происходит до тех пор, пока частное не станет меньше основания системы счисления.  В результате  X = aN a(N-1)…a1 a0 ,  есть представление в k-ичной системе счисления. |

|  |  |
| --- | --- |
| **2. Пример** | |
| ***Задача:*** | |
|  | Дано: 4810 = X3. Найти X. |
| ***Решение:*** | |
|  | 4810 делим на 3, частное = 16, остаток a0 =  частное = 1610 делим на 3, частное = 5, остаток a1 = 1  частное = 510 делим на 3, частное = 1, остаток a2 = 2  частное = 110 делим на 3, частное = 0, остаток a3 = 1  Частное не больше нуля, деление закончено. Для представления числа в заданной системе счисления остатки от деления записываются в обратном порядке:  4810= (a3a2a1a0)3 = 12103. |
| ***Ответ:*** | |
|  | 4810 = 12103. |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Дано: 36710 = X7. Найти X. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 1033 |
| 2. | ***Задача:*** | |
|  | Дано: 114310 = X12. Найти X. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 7b3 |

|  |  |  |
| --- | --- | --- |
| 3. | ***Задача:*** | |
|  | Дано: 1278 = X9. Найти X. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 151 |
| 4. | ***Задача:*** | |
|  | Дано: AB413 = X6. Найти X. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 114454 |

|  |
| --- |
| **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 | | 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 системы 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 10111011102 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. **10111011102 = 13568** |
| ***Ответ:*** | |
|  | **1356** |
| ***Задача:*** | |
|  | Перевести шестнадцатеричное число 3AC16 в двоичную систему счисления. |
| ***Решение:*** | |
|  | Используем алгоритм, приведенный выше:   1. **0011.1010.1100** 2. **001110101100** 3. **1110101100**   Таким образом, 3AC16 = 11101011002 |
| ***Ответ:*** | |
|  | **1110101100** |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Дано: 10100111012 = X16. Найти X. |
| ***Решение:*** | |
|  | # bin(2) 1010011101 -> hex(16)  # 0010.1001.1101 -> 2 9 d |
| ***Ответ:*** | |
|  | 29d |
| 2. | ***Задача:*** | |
|  | Дано: 11474538 = X2. Найти X. |
| ***Решение:*** | |
|  | # oct(8) 1147453 -> bin(2)  # 1 1 100 111 100 101 11  # 001 001 100 111 100 101 011 |
| ***Ответ:*** | |
|  | 001001100111100101011 |
| 3. | ***Задача:*** | |
|  | Дано: 1278 = X16. Найти X. |
| ***Решение:*** | |
|  | # oct(8) 127 -> hex(16)  # 127 -> 1 10 111  # 0001 0010 0111 -> 1 2 7 |
| ***Ответ:*** | |
|  | 127 |

|  |  |  |
| --- | --- | --- |
| 4. | ***Задача:*** | |
|  | Сколько значащих нулей в двоичной записи шестнадцатеричного числа C3E116? |
| ***Решение:*** | |
|  | # hex(16) c3e1 -> bin(2)  # 1100 0011 1110 0001 |
| ***Ответ:*** | |
|  | 8 |
| 5. | ***Задача\*:*** | |
|  | Дано: AF3832 = X16 = X8 = X2. Найти X. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 53c68(16) = 1236150(8) = 01010011110001101000(2) |
| 6. | ***Задача\*:*** | |
|  | 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 \*  X = E\*16 = \*5\*8 = \*\*\*14 = \*\*\*\*\*1\*\*2  Определите число X. |
| ***Решение:*** | |
|  |  |
| ***Ответ:237*** | |
|  | 237 |

|  |  |  |
| --- | --- | --- |
| **Тест 2** | | |
| **1.** | ***Задание:*** | |
|  | Переведите число 37 из десятичной системы счисления в двоичную: |
| ***Ответ:*** | |
|  | А) 100101 |
| **2.** | ***Задание:*** | |
|  | Переведите число 110102 из двоичной системы счисления в десятичную систему счисления. |
| ***Ответ:*** | |
|  | С) 26 |

|  |  |  |
| --- | --- | --- |
| **3.** | ***Задание:*** | |
|  | Дано: а = D716, b = 3318. Какое из чисел с, записанных в двоичной системе, отвечают условию a<c<b? |
| ***Ответ:*** | |
|  | D) 11011000 |
| **4.** | ***Задание:*** | |
|  | Для чисел, заданных в различных системах счисления: X = 11123, Y = 1405, Z = 2224 – справедливо соотношение: |
| ***Ответ:*** | |
|  | B) X<Z<Y |
| **5.** | ***Задание:*** | |
|  | В системе счисления с некоторым основанием десятичное число 144 записывается в виде 264. Укажите это основание. |
| ***Ответ:*** | |
|  | C) 7 |
| **6.** | ***Задание:*** | |
|  | Укажите основание системы счисления, в которой не может быть записано число 1302 |
| ***Ответ:*** | |
|  | A) 3 |
| **7.** | ***Задание:*** | |
|  | Значение выражения 1016 + 108 \* 102 в двоичной системе счисления равно |
| ***Ответ:*** | |
|  | C) 100000 |
| **8.** | ***Задание:*** | |
|  | Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 40 оканчивается на 4. |
| ***Ответ:*** | |
|  | **6, 9, 12, 18, 36** |
| **9.** | ***Задание:*** | |
|  | В системе счисления с некоторым основанием десятичное число **52** записывается в виде **202n**. Найдите **n** . |
| ***Ответ:*** | |
|  | **5** |

|  |  |  |
| --- | --- | --- |
| **10.** | ***Задание:*** | |
|  | Решите уравнение: 126 + *x* = 3245 Ответ запишите в десятичной системе счисления. |
| ***Ответ:*** | |
|  | **81** |

Реализация задач на языке программирования Python

|  |  |
| --- | --- |
| **1. Теоретический материал** | |
| Для перевода числа из одной системы счисления в другую в Python существует несколько функций:   * **int([строка], [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(х)** - преобразование целого числа в шестнадцатеричную строку. * **oct(х)** - 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):*** | |
|  | print('Введите число в десячиной системе счисления')  a = int(input())  print('Двоичная: ', bin(a))  print('Восьмеричная: ', oct(a))  print('Шестнадцатиричная: ', hex(a)) |

|  |  |
| --- | --- |
| **3. Задания** | |
| ***Задача:*** | |
|  | На вход программа получает две величины: *n, A,* где *n* – натуральное числа от 2 до 36, основание системы счисления, *A* – число, записанное в системе счисления с основанием *n*, *A* < 231.  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):*** | |
|  |  |

|  |
| --- |
| 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**: четырьмя 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 х')) # 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:  print("Третья четверть") |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Дано двузначное число. Определить входит ли в него цифра 3. (// - операция получения целой части от деления, % - операция взятия остатка от целочисленного деления). |
| ***Решение (код программы):*** | |
|  |  |
| 2. | ***Задача:*** | |
|  | Дано двузначное число. Определить какая из его цифр больше. |
| ***Решение (код программы):*** | |
|  |  |
| 3. | ***Задача:*** | |
|  | 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. |
| ***Решение (код программы):*** | |
|  |  |

|  |
| --- |
| 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](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)).  [Выражение](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%B2%D1%8B%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), 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 функций **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 встречает число 5. |
| ***Решение (код программы):*** | |
|  | n = [1, 2, 3, 7, 6, 4, 5, 8] #пример списка  for x in n:  if x == 5:  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 = '' не переводит на новую строку |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1. |
| ***Решение (код программы):*** | |
|  |  |
| 2. | ***Задача:*** | |
|  | Посчитать сумму числового ряда от 1 до *N* включительно (т.е. 0+1+2+3+…+ *N*). Решите задачу с помощью и без помощи оператора цикла. Число *N* вводится с клавиатуры (*N*< 1000). |
| ***Решение (код программы):*** | |
|  |  |
| 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):*** | |
|  |  |
| 4. | ***Задача:*** | |
|  | Вводится десятичное число *A* (*A*< 231) и число *n* (2 ≤ *n* ≤ 9). Необходимо перевести введенное число *A* в систему счисления c основанием *n*. При этом использовать встроенные конструкции языка Python не разрешается. |
| ***Решение (код программы):*** | |
|  |  |