**ТЕМА 2**

**Работа с целыми числами**

1. **Работа с цифрами числа**
	1. **Арифметика работы с цифрами целого числа.**
2. Целые числа можно разложить по степеням 10. Например число 2371 раскладывается в 2371 = 2\*103 + 3\*102 + 7\*101 + 1\*100 ( обычно мы это записываем проще 2371 = 2\*103 + 3\*102 + 7\*10 + 1. Здесь 2, 3, 7, 1 являются цифрами нашего числа. Мы будем в дальнейшем иногда изображать целое число N в виде N = edcba , где **a, b, c, d** являются цифрами числа ( в данном случае пятизначного числа).
3. Если нам даны цифры c – количество сотен, b – количество десятков, a – количество единиц, то они формируют число N = 100\*с + 10\*b + a. Например, если даны цифры числа **a=4, b=6, c=2** и мы хотим получить из них число **N=462**, то **N = 100\* c + 10\*b + a.**
4. Добавить, приписать, справа цифру **a**  к числу **N: N = N\*10 + a.** Например, если **N = 34, a=7, N\*10 + a = 347,**  то есть мы к числу 34 справа приписали цифру 7.
5. Для выделения и удаления цифр в числе в Python соответственно используются операции “%” и “//” - целое деление.

Внимание. Выделять и удалять цифры мы будем только у положительных чисел. (см. 2.1 Tema 1)

 Примеры выделения и удаления цифр.

1. 72396%10 = 6: 72396%100 = 96: 72396%1000 = 3966: 72396%10000 = 2396:
2. 72396 //10 = 7239: 72396 //100 = 723: 72396 //1000 = 72: 72396 //10000 = 7:
3. Пусть дано целое число N = 458932.

|  |  |
| --- | --- |
| Что сделать | Python |
| Выделить 1-ую цифру справа | a = N % 10 |
| Выделить 2-ую цифру справа | b = N // 10 % 10 |
| Выделить 3-ую цифру справа | c = N // 100 % 10 |
| Выделить 4-ую цифру справаТо есть удалить 3 цифры и выделить четвертую | d = N // 1000 % 10 |
| Выделить 5-ую цифру справа | c = N // 10000 % 10 |
| Выделить 6-ую цифру справа | d = N // 100000 % 10 |
| Так число точно 6-ти значное, тоМожно и так: d = N // 100000 | d = N // 100000%10 |

1. Как мы отметили в п. 4, операция “%” – нахождение остатка от деления ( не только на 10, как нам было нужно в п. 4). Например, 10%3=1, 15%4=3, 14%3=2, 15%3=0. Если остаток от деления числа **a** на **b** равен 0, то это значит, что число **a** делится без остатка на **b**, то есть **b** является делителем **a** или, как мы говорим, **a** нацело делится на **b**. Например, если **a%2=0**, то это значит, что **a** без остатка делится на 2, то есть **a** является четным числом. Это и есть проверка на четность числа.
	1. Работа с целыми числами в виде текста.
2. Пусть дано число s = '123'( в ввиде строки), то каждую цифру (как текстовой символ) можно выделит таким образом: u = s[0], v = s[1], w = s[2], а если мы хотим получить числовые цифры, то выполним: a = int(s[0]), b = int(s[1]), c = int(s[2]).

Само число мы можем получить:

1. N = 100\*a + 10\*b + c
2. N = int(u + v + w)
3. Пусть дано число s = '123' ( в ввиде строки). Преобразуем s в список и мы получим:

>>> s1=list(s)

>>> print(s1)

['1', '2', '3']

Теперь можем работать с отдельными элементами списка s1 и помним, что список s1 можно менять (добавлять и удалять элементы), а строку s менять нельзя (то есть строки неизменяемые данные)

**Необходимые дополнительные сведения о списках в Python**

Пусть дан список s = [7, 3, -4, 4, 8]

|  |  |  |
| --- | --- | --- |
|  |  | Примеры |
| **s.append**(x) | Добавляет элемент в конец списка | s = [7, 3, -4, 4, 8]s.append(100)print(s)[7, 3, -4, 4, 8, 100] |
| **s.extend**(L) | Расширяет список list, добавляя в конец все элементы списка L | s = [7, 3, -4, 4, 8]s1=[200,300,400]s.extend(s1)print(s)[7, 3, -4, 4, 8, 200, 300, 400] |
| **s.insert**(i, x) | Вставляет на i-ый элемент значение x | s = [7, 3, -4, 4, 8]s.insert(3, -100)print(s)[7, 3, -4, -100, 4, 8] |
| **s.remove**(x) | Удаляет первый элемент в списке, имеющий значение x. ValueError, если такого элемента не существует | s = [7, 3, -4, 4, 8, -4, 3]s.remove(-4)print(s)[7, 3, 4, 8, -4, 3] |
| **s.pop**([i]) | Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент | s = [7, 3, -4, 4, 8, -4, 3]a=s.pop(2)print(s)print(a)[7, 3, 4, 8, -4, 3]-4 |
| **s.index**(x, [start [, end]]) | Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end) | s = [7,-4, 3, -4, 4, 8, -4, 3]print(s.index(-4,2,7)3print(s.index(-4,4)6 |
| **s.count**(x) | Возвращает количество элементов со значением x | s = [7, 3, -4, 4, 8, -4, 3]k=s.count(-4)print(k)2 |
| **s.sort**() | Сортирует сам список по возрастанию | s = [7, 3, -4, 4, 8, -4, 3]s.sort()print(s)[-4, -4, 3, 3, 4, 7, 8] |
| **s.sort(reverse = True)** | Сортирует сам список по убыванию | s = [7, 3, -4, 4, 8, -4, 3]s.sort(reverse = True)print(s)[8, 7, 4, 3, 3, -4, -4] |
| **s.sorted**() | Возвращает отсортированый по возрастанию список.Исходный список не изменяется | s = [7, 3, -4, 4, 8, -4, 3]s1=s.sort()print(s)[7, 3, -4, 4, 8, -4, 3]print(s1)[-4, -4, 3, 3, 4, 7, 8] |
| **s.sorted(reverse = True)** | Возвращает отсортированый по возрастанию список.Исходный список не изменяется | s = [7, 3, -4, 4, 8, -4, 3]s1=s.sort(reverse = True)print(s)[7, 3, -4, 4, 8, -4, 3]print(s1)[8, 7, 4, 3, 3, -4, -4] |
| **s.reverse**() | Разворачивает список | s = [7, 3, -4, 4, 8, -4, 3]s.reverse()print(s)[3, -4, 8, 4, -4, 3, 7] |
| **s.copy**() | Копия списка | s = [7, 3, -4, 4, 8]s1=[200,300,400]s1=s.copy()print(s1)[7, 3, -4, 4, 8] |
| **s.clear**() | Очищает список | s = [7, 3, -4, 4, 8, -4, 3]s.clear()print(s)[] |
| **max(s), min(s)**  | Находит максимальный, минимальный элементы | s = [7, 3, -4, 4, 8, -4, 3]print(max(s),min(s))8 -4 |
| **sum(s)** | Находит сумму элементов | s = [7, 3, -4, 14]print(sum(s))20 |

1. **Примеры работы с цифрами целого числа**

В примерах ниже мы будем приводит разные решения одной и той же задачи: работа с числами, как целыми числами, как строками, как списками.

* 1. Дано трехзначное натуральное число. Найти сумму его цифр.

В этой первой задаче приведем разные варианты записей решения.

|  |  |  |  |
| --- | --- | --- | --- |
| № |  | Решение | Пояснение  |
| 1 | 123456 | n=int(input())a= n//100%10b=n//10%10c= n%10s=a+b+cprint(s) | Ввод числавыделение первой цифры слевавыделение второй цифры слевавыделение последней цифрывычисление суммы цифрвывод на экран ответа |
| 123 | n=int(input())a,b,c=n//100, n//10%10, n%10print(a+b+c) | Это та же программа, что и наверху, только строки 2,3,4 объединены в одну строку 2 в этой программе |
| **2** | 123 | s=input()a,b,c=int(s[0]),int(s[1]),int(s[2])print(a+b+c) | Число вводится в текстовом видеКаждая цифра переводится в цисловывод на экран ответа |
| 12 | s=input()print(int(s[0])+int(s[1])+int(s[2])) | Та же самая программа, но без дополнительных переменных |
| **3** | 123 | s=list(input())s=[int(a) for a in s]print(sum(s)) | list преобразуется число в список цифр в символьном виде – ‘0’, ‘1’, ‘2’,…’9’Символьные цифры преобразуются в числаsum(s) вычисляет сумму цифр |
| 1 | print(sum([int(a) for a in input()])) | Та же программа в одной строке |
| **Если в каждой из этих 6 программ ввести число 245, то получим 11**  |

* 1. Дано натуральное число. Найти максимальную цифру в этом числе.

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
| **1** | n=int(input())maxd=0while n>0: q=n%10 if q>maxd: maxd=q n=n//10print(maxd) | Ввод числав maxd будем хранить максимальную цифруВыделяем и удаляем цифры пока n>0Выделяем последню цифру числа nЕсли цифра больше максимальноймаксимальную заменяем на эту цифруудаляем последнюю цифрувывод ответа |
| **2** | s=list(input())maxd=int(max(s))print(maxd) | list преобразуется число в список цифр в символьном виде – ‘0’, ‘1’, ‘2’,…’9’max(s) находит максимальную цифрувывод ответа |
| print(int(max(list(input())))) | Тоже в одной строке |
| **Если в каждой из этих 3 программ ввести число 2454032, то получим 5** |

* 1. Дано натуральное число. Найти количество максимальных цифр в этом числе.

Вывести максимальную цифру и число ее повторений. Например, в числе 273527347536 максимальная цифра 7 и она повторяется 3 раза.

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  **1** | n=int(input())maxd=0k=1while n>0: q=n%10 if q>maxd: maxd=q k=1 else: if q==maxd: k=k+1 n=n//10print(maxd,k) | Ввод числав maxd храним максимальную цифру, а в k – количество макс. цифрвыделяем и удаляем цифры пока n>0выделяем последнюю цифру числа nЕсли цифра больше максимальной, то maxd становится равной этой цифреи k=1, то есть maxd снова однаиначеесли maxd равна цифре, тоувеличиваем kудаляем последнюю цифрувывод ответа |
|  **2** | n= list(input())maxd=max(n)k=n.count(maxd)print(int(maxd), k) | list преобразуется число в список цифр в символьном виде – ‘0’, ‘1’, ‘2’,…’9’max(n) находит максимальную цифруn.count(maxd) находит количество цифр вывод ответа |
| **Если в этих 2 программ ввести число 273527347536, то получим 7 3** |

* 1. Дано натуральное число N (N>=1000). В этом числе поменять местами число тысяч и число десятков. Например, число 2341**5**6**7**4 преобразуется в число 2341**7**6**5**4

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  **1** | n=int(input())m=n//10000a=n//1000%10b=n//100%10c=n//10%10d=n%10x=m\*10000+c\*1000+b\*100+a\*10+dprint(x) | Ввод числа – например 2341**5**6**7**4удаляем последние 4 цифры и получаем 2341выделяем 5выделяем 6выделяем 7выделяем 4содираем новое число2341\*10000+7\*1000+6\*100+5\*10+4вывод ответа |
| n=int(input())m,a=n//10000,n//1000%10b,c,d=n//100%10,n//10%10,n%10print((((m\*10+c)\*10+b)\*10+a)\*10+d) | Это таже самая программа, только объединили операторы |
|  **2** | n=input()l=len(n)m=n[:l-4]+n[l-2]+n[l-3]+n[l-4]+n[l-1]print(int(m)) | Ввод числа в текстовом видеопределения длины числаформирование нового текстового числаиз цифртекстовое число переводим в целоеи выводим на экран  |
|  **3** | n=list(input())l=len(n)n[l-4],n[l-2]=n[l-2],n[l-4]m=''.join(n)print(int(m)) | list преобразуется число в список цифр в символьном виде – ‘0’, ‘1’, ‘2’,…’9’определения количество цифр числаменяем местами 5 и 7(2341**5**6**7**4)''.join(n) из списка цифр создает число в текстовом видетекстовое число переводим в целоеи выводим на экран |

* 1. Найти количество натуральных чисел, не превосходящих заданного N (N<100000), и делящихся на каждую из своих цифр.

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  **1** | n=int(input())k=0for i in range(1,n+1): m=i p=0 while m>0: q=m%10 if q==0 or i%q!=0: p=1 break m=m//10 if p==0: k+=1print(k) | проссматриваем числа от 1 до n включителнодо проверки p=0каждое число проверяем равна ли цифра нулю или исходное число не делится на цифру.Если это так, то p=1 и п выходим из цикла whileесли же мы не вышли из цикло раньше по break, то есть p осталось равным 0, увеличиваем k |
|  **2** | n=int(input())k=0for i in range(1,n+1): m=i; p=0 while m>0: q=m%10 if q==0 or i%q!=0: p=1; break m=m//10 if p==0: k+=1print(k) | Та же программа, просто показываем, что некоторые операторы можно записываит в одну строчка |

* 1. Дано натуральное число N. Удалить из этого числа все нечетные цифры, оставив порядок цифр прежним. Например, число 527014 преобразуется в 204.

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  **1** | n=int(input())m=0; p=1while n>0: q=n%10 if q%2==0: m=p\*q+m p=p\*10 n=n//10print(m) | проверяем каждую цифруесли она четная, то добавляем ее к m с слева по формуле m=p\*q+m, где p это спепень 10 для добавления цифры слева |
|  **2** | s=input()l=len(s)n=''for i in range(l): if int(s[i])%2==0: n=n+s[i]print(int(n)) | проссматриваем все цифры слева направо и если она четнаядобавляем к тексту n |
|  **3** | s=list(input())n=[a for a in s if int(a)%2==0]print(int(''.join(n))) | список цифрсоздаем новый список только из четных цифр''.join(n) из списка цифр создает число в текстовом видетекстовое число переводим в целоеи выводим на экран |
| **Если в каждой из этих 3 программ ввести число 527014, то получим 204** |

* 1. Дано натуральное число N. Переставить цифры этого числа так, чтобы получилось наибольше возможное число. Например: из числа 25317623 надо получить 76533221.

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  | s= list(input())s=sorted(s,reverse=True)n=int(''.join(s))print(n) | сортировка списка цифр по убыванию |
| print(int(''.join(sorted(list(input()),reverse=True)))) | то же в одной строке |

* 1. Даны натуральные числа m < n (100<=m,n<=999). Найти все трехзначные числа Амстронга в промежутке от m до n включительно. Трехзначное число является числом Амстронга, если сумма кубов цифр этогочисла равна самому числу. Например: 153 = 13 + 53 + 33 .

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  | a,b=map(int, input().split())for i in range(a,b+1): a,b,c=i//100, i//10%10, i%10 if a\*\*3+b\*\*3+c\*\*3 == i:  print(i) | ввод двух чисел в одной строкепроверяем все числа от a до b включительновыделяем цифры ели сумма кубов цифр равна самому числу (i),выводим его |

* 1. Найти все N – значные натуральные числа, сумма цифр каждого из которых равна заданному k*.*

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  **1** | n,k=map(int, input().split())a=10\*\*(n-1)for i in range(a,10\*a): s=0 m=i while m>0: s = s + m%10 m=m//10 if s==k: print(i,end=' ') | проверяем все n-значные числанаходим сумму цифр числасумму сравниваем с k и,если сумма равна k, тот выводим числа (i) в строку (end=' ') |
|  **2** | n,k=map(int, input().split())a=10\*\*(n-1)for i in range(a,10\*a): s=[int(b) for b in list(str(i))] if sum(s)==k: print(i,end=' ') | число переводим в строку цифрsum(s) – сумма цифр |
| **Если ввести числа 3 5, то получим 104 113 122 131… 500** |

|  |  |  |
| --- | --- | --- |
| № | Решение | Пояснение  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |