

Строки и символы

Коды символов

Любой текст состоит из символов. Символ — это некоторый значок, изображение. Один и тот же символ можно записать по-разному, например, два человека по-разному напишут от руки букву “А”, и даже в компьютерном представлении одна и та же буква будет выглядеть по-разному, если ее отображать разными шрифтами, при этом это будет все равно один и тот же символ. Верно и другое: разные символы могут быть записаны одинаково, например, вот две разные буквы, одна — латинского алфавита, другая - русского: “А” и “А”. Несмотря на то, что они выглядят одинаково, удобней считать их разными символами.

Итак, способ хранения текстовой информации в компьютере не связан напрямую с изображением этого текста. Вместо символов хранятся их номера - числовые коды, а вот то, как выглядит символ с данным числовым кодом на экране напрямую зависит от того, какой используется шрифт для отображения символов. При этом, разумеется, следовало бы договориться о единообразном способе кодирования символов числовыми кодами, иначе текст, записанный на одном компьютере, невозможно будет прочесть на другом компьютере.

Первоначально договорились под кодирование одного символа отвести один байт, то есть 8 бит информации. Таким образом можно было закодировать 256 различных значений, то есть в записи текста можно использовать 256 различных символов. Этого достаточно, чтобы отобразить все символы латинского алфавита, цифры, знаки препинания и некоторые другие символы. Стандарт, указывающий, какие числовые коды соответствуют каким основным символам, называется [ASCII](#). В таблицу ASCII включены символы с кодами от 0 до 127, то есть ASCII - это семибитный код. Вот так выглядит таблица ASCII:

Char Dec Oct Hex | Char Dec Oct Hex | Char Dec Oct Hex | Char Dec Oct Hex

```
-----
(nul) 0 0000 0x00 | (sp) 32 0040 0x20 | @ 64 0100 0x40 | ` 96 0140 0x60
(soh) 1 0001 0x01 | ! 33 0041 0x21 | A 65 0101 0x41 | a 97 0141 0x61
(stx) 2 0002 0x02 | " 34 0042 0x22 | B 66 0102 0x42 | b 98 0142 0x62
(etx) 3 0003 0x03 | # 35 0043 0x23 | C 67 0103 0x43 | c 99 0143 0x63
(eot) 4 0004 0x04 | $ 36 0044 0x24 | D 68 0104 0x44 | d 100 0144 0x64
(enq) 5 0005 0x05 | % 37 0045 0x25 | E 69 0105 0x45 | e 101 0145 0x65
(ack) 6 0006 0x06 | & 38 0046 0x26 | F 70 0106 0x46 | f 102 0146 0x66
(bel) 7 0007 0x07 | ' 39 0047 0x27 | G 71 0107 0x47 | g 103 0147 0x67
(bs) 8 0010 0x08 | ( 40 0050 0x28 | H 72 0110 0x48 | h 104 0150 0x68
(ht) 9 0011 0x09 | ) 41 0051 0x29 | I 73 0111 0x49 | i 105 0151 0x69
(nl) 10 0012 0x0a | * 42 0052 0x2a | J 74 0112 0x4a | j 106 0152 0x6a
(vt) 11 0013 0x0b | + 43 0053 0x2b | K 75 0113 0x4b | k 107 0153 0x6b
( Loading [MathJax]/extensions/MathZoom.js 76 0114 0x4c | l 108 0154 0x6c
(cr) 13 0015 0x0d | - 45 0055 0x2d | M 77 0115 0x4d | m 109 0155 0x6d
```

| | | | | | | | | | | | | | | | |
|-------|----|------|------|---|----|------|------|---|----|------|------|-------|-----|------|------|
| (so) | 14 | 0016 | 0x0e | . | 46 | 0056 | 0x2e | N | 78 | 0116 | 0x4e | n | 110 | 0156 | 0x6e |
| (si) | 15 | 0017 | 0x0f | / | 47 | 0057 | 0x2f | O | 79 | 0117 | 0x4f | o | 111 | 0157 | 0x6f |
| (dle) | 16 | 0020 | 0x10 | 0 | 48 | 0060 | 0x30 | P | 80 | 0120 | 0x50 | p | 112 | 0160 | 0x70 |
| (dc1) | 17 | 0021 | 0x11 | 1 | 49 | 0061 | 0x31 | Q | 81 | 0121 | 0x51 | q | 113 | 0161 | 0x71 |
| (dc2) | 18 | 0022 | 0x12 | 2 | 50 | 0062 | 0x32 | R | 82 | 0122 | 0x52 | r | 114 | 0162 | 0x72 |
| (dc3) | 19 | 0023 | 0x13 | 3 | 51 | 0063 | 0x33 | S | 83 | 0123 | 0x53 | s | 115 | 0163 | 0x73 |
| (dc4) | 20 | 0024 | 0x14 | 4 | 52 | 0064 | 0x34 | T | 84 | 0124 | 0x54 | t | 116 | 0164 | 0x74 |
| (nak) | 21 | 0025 | 0x15 | 5 | 53 | 0065 | 0x35 | U | 85 | 0125 | 0x55 | u | 117 | 0165 | 0x75 |
| (syn) | 22 | 0026 | 0x16 | 6 | 54 | 0066 | 0x36 | V | 86 | 0126 | 0x56 | v | 118 | 0166 | 0x76 |
| (etb) | 23 | 0027 | 0x17 | 7 | 55 | 0067 | 0x37 | W | 87 | 0127 | 0x57 | w | 119 | 0167 | 0x77 |
| (can) | 24 | 0030 | 0x18 | 8 | 56 | 0070 | 0x38 | X | 88 | 0130 | 0x58 | x | 120 | 0170 | 0x78 |
| (em) | 25 | 0031 | 0x19 | 9 | 57 | 0071 | 0x39 | Y | 89 | 0131 | 0x59 | y | 121 | 0171 | 0x79 |
| (sub) | 26 | 0032 | 0x1a | : | 58 | 0072 | 0x3a | Z | 90 | 0132 | 0x5a | z | 122 | 0172 | 0x7a |
| (esc) | 27 | 0033 | 0x1b | ; | 59 | 0073 | 0x3b | [| 91 | 0133 | 0x5b | { | 123 | 0173 | 0x7b |
| (fs) | 28 | 0034 | 0x1c | < | 60 | 0074 | 0x3c | \ | 92 | 0134 | 0x5c | | 124 | 0174 | 0x7c |
| (gs) | 29 | 0035 | 0x1d | = | 61 | 0075 | 0x3d |] | 93 | 0135 | 0x5d | } | 125 | 0175 | 0x7d |
| (rs) | 30 | 0036 | 0x1e | > | 62 | 0076 | 0x3e | ^ | 94 | 0136 | 0x5e | ~ | 126 | 0176 | 0x7e |
| (us) | 31 | 0037 | 0x1f | ? | 63 | 0077 | 0x3f | _ | 95 | 0137 | 0x5f | (del) | 127 | 0177 | 0x7f |

При этом символы с кодами, меньшими 32 - это специальные управляющие символы, которые не отображаются на экране. Например, для того, чтобы обозначить конец строки в системе Linux используется один символ с кодом 10, а в системе Windows - два подряд идущих символа с кодами 13 и 10, символы с кодами 48-57 соответствуют начертанию арабских цифр (обратите внимание, символ с кодом 0 - это вовсе не символ, отображающийся на экране, как "0"), символы с кодами 65-90 - заглавные буквы латинского алфавита, а если к их кодам прибавить 32, то получатся строчные буквы латинского алфавита. В промежутках между указанными диапазонами находятся знаки препинания, математические операции и прочие символы.

Но в ASCII-таблицы нет русских букв! А также нет букв сотен других национальных алфавитов. Первоначально для отображения букв национальных алфавитов использовали вторую половину возможного значения байта, то есть символы с кодами от 128 до 255. Это приводило к множеству проблем, например, поскольку 128 значений явно недостаточно для того, чтобы отобразить символы всех национальных алфавитов (даже недостаточно для того, чтобы отобразить символы одного алфавита, например, [китайской письменности](#)). Поэтому в настоящее время для кодирования символов используется стандарт [Unicode](#), последняя версия 6.0 которого (октябрь, 2010) включает свыше 109000 различных символов. Естественно, для кодирования Unicode-символов недостаточно одного байта на символ, поэтому используются многобайтовые кодировки (для представления одного символа необходимо несколько байт).

Язык программирования Python — современный язык, поэтому он работает исключительно с Unicode-символами.

Код символа можно определить при помощи функции `ord`. Эта функция получает на вход строку, состоящую из одного символа. Функция возвращает код этого символа.

Loading [MathJax]/extensions/MathZoom.js

Например, `ord('A')` вернет число 65.

Обратная функция получения по числовому коду его номера называется `chr`.

Сравнение символов

Поскольку для символов заданы их числовые коды, то их можно сравнивать при помощи операций сравнения. Поскольку символы алфавита идут подряд, то результат их сравнения будет соответствовать лексикографическому порядку, но можно сравнивать между собой не только буквы алфавита, но и два произвольных символа.

Также в питоне определены и операции сравнения строк, которые также сравниваются в лексикографическом порядке.

Строки и списки

Строки в языке Питон, в отличие от списков, являются неизменными объектами, то есть в отличие от списков нельзя изменить отдельный символ строки: операция `A[i] = ...` является допустимой, если `A` — список, и недопустимой, если `A` — строка.

Один символ в строке можно поменять, если создать новую строку, например, следующий код меняет `i`-й символ строки на символ `@`.

```
A = A[:i] + '@' + A[i+1:]
```

К сожалению, подобное изменение требует время, пропорциональное длине строки.

Преобразование строки `S` в список символов `A`:

```
A = list(S)
```

Преобразование списка символов в строку:

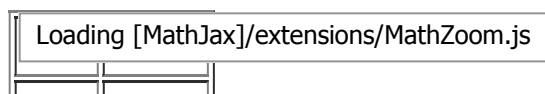
```
S = ''.join(A)
```

Упражнения

A: ASCII-код символа

Считайте со стандартного ввода символ и выведите его код.

Программа получает на вход один символ с кодом от 33 до 126.



| | |
|---|----|
| A | 65 |
|---|----|

B: Символ с данным ASCII-кодом

Считайте со стандартного ввода целое число и выведите ASCII-символ с таким кодом. Решите эту задачу с использованием только одной переменной типа `int`.

Программа получает на вход число от 33 до 126.

| Ввод | Вывод |
|------|-------|
| 65 | A |

C: Таблица ASCII

Выведите все символы ASCII с кодами от 33 до 126 и их коды в следующем виде:

| Вывод |
|-------|
| ! 33 |
| " 34 |
| # 35 |
| ... |
| } 125 |
| ~ 126 |

D: Символы в заданном интервале

Выведите подряд, без пробелов, все символы, лежащие в таблице ASCII между двумя заданными символами.

Программа получает на вход два символа, каждый в отдельной строке и должна вывести строку, начинающуюся первым из заданных символов и заканчивающуюся вторым.

| Ввод | Вывод |
|--------|------------|
| A D | ABCD |
| 0 9 | 0123456789 |

Loading [MathJax]/extensions/MathZoom.js

E: IsDigit

Для данного символа, считанного со стандартного ввода, проверьте, является ли он цифрой. Программа должна вывести слово YES, если символ является цифрой, или слово NO.

Решение оформите в виде функции IsDigit(c), возвращающей значение типа bool. В решении нельзя использовать циклы. В решении нельзя использовать константы с неочевидным значением типа 48 или 57.

| Ввод | Вывод |
|------|-------|
| 0 | YES |
| A | NO |

F: ToUpper

Напишите функцию ToUpper(c), которая переводит символ в верхний регистр, то есть для строчной буквы латинского алфавита возвращает соответствующую заглавную букву латинского алфавита, а для остальных символов возвращает тот же символ.

Считайте один символ со стандартного ввода и переведите его в верхний регистр. В решении нельзя использовать циклы. В решении нельзя использовать константы с неочевидным значением.

| Ввод | Вывод |
|------|-------|
| f | F |
| F | F |
| 4 | 4 |

G: Сменить регистр символа

Напишите функцию CaseChange (c), меняющую регистр символа, то есть переводящую заглавные буквы в строчные, а строчные - в заглавные, остальные символы не меняющие.

Считайте один символ со стандартного ввода, выведите результат работы данной функции. В решении нельзя использовать константы с неочевидным значением.

Loading [MathJax]/extensions/MathZoom.js

| Ввод | Вывод |
|------|-------|
| f | F |
| F | f |
| 4 | 4 |

Н: Нижний регистр

Дана строка, возможно, содержащая пробелы. Считайте эту строку и переведите все символы этой строки в нижний регистр. Решение оформите в виде функции `ToLower(S)`, получающей в качестве параметра строку и возвращающую новую строку.

Для перевода одного символа в нижний регистр напишите отдельную функцию.

| Ввод | Вывод |
|---------------|---------------|
| Hello, world! | hello, world! |

І: Палиндром

Дано слово, состоящее только из заглавных и строчных латинских букв. Проверьте, верно ли что это слово читается одинаково как справа налево, так и слева направо (то есть является палиндромом), если считать заглавные и строчные буквы не различающимися. Выведите слово YES, если слово является палиндромом и словов NO, если не является.

Решение оформите в виде функции `IsPalindrome(S)`, возвращающей значение типа `bool`.

Решение должно иметь сложность $O(n)$, где n — длина строки.

В решении задачи нельзя использовать срез для разворота строки (а также `reverse/reversed`). Используйте алгоритм **линейного поиска**, то есть цикл `while` без `if` внутри.

| Ввод | Вывод |
|-------|-------|
| Radar | YES |
| YES | NO |

Loading [MathJax]/extensions/MathZoom.js

Ж: Извлечь цифры

Дана строка, возможно, содержащая пробелы. Извлеките из этой строки все символы, являющиеся цифрами и составьте из них новую строку. Решение оформите в виде функции `ExtractDigits(S)`, получающей на вход исходную строку `S` и возвращающую новую строку, содержащую только цифры данной строки.

| Ввод | Вывод |
|-------|-------|
| 2+2=4 | 224 |

К: Значение выражения - 1

Дана строка, состоящая из n цифр (т.е. однозначных чисел), между которыми стоит $n-1$ знак операции, каждый из которых может быть либо $+$, либо $-$. Вычислите значение данного выражения.

Решение оформите в виде функции `Evaluate(S)`, получающей в качестве аргумента строку и возвращающую ее значение.

| Ввод | Вывод |
|-------|-------|
| 1+2-3 | 0 |

Л: Значение выражения - 2

Дано выражение одно из следующих видов: " $A+B$ ", " $A-B$ " или " $A*B$ ", где A и B - целые неотрицательные числа. Определите значение этого выражения.

Решение оформите в виде функции `Eval(S)`.

| Ввод | Вывод |
|---------|-------|
| 2*2 | 4 |
| 100-101 | -1 |

М: Шифр Цезаря

Каждый символ заданной строки заменяется на другой символ, третий по счету в алфавите после данного, с цикличностью. То есть символ `A` заменяется на `D`, символ `B` - на `E`, символ `C` - на

F, ..., символ Z на C. Аналогично строчные буквы заменяются на строчные буквы. Все остальные символы не меняются.

Дана строка, зашифруйте ее при помощи шифра Цезаря. Решение оформите в виде функции CaesarCipher (S, k), возвращающей новую строку. S — исходная строка, k — величина сдвига. Функцию нужно вызывать со значением k=3.

Указание: сделайте функцию CaesarCipherChar(c, k), шифрующую один символ.

| Ввод | Вывод |
|---|---|
| In a hole in the ground there lived a hobbit. | Lq d kroh lq wkh jurxqg wkhuh olyhg d kreelw. |

N: Количество слов

Дана строка, возможно, содержащая пробелы. Определите количество слов в этой строке. Слово — это несколько подряд идущих букв латинского алфавита (как заглавных, так и строчных).

Решение оформите в виде функции CountWords (S), возвращающее значение типа int. При решении этой задачи нельзя пользоваться дополнительными строками и списками.

| Ввод | Вывод |
|---|-------|
| Yesterday, all my troubles seemed so far away | 8 |

O: Самое длинное слово

Дана строка. Найдите в этой строке самое длинное слово и выведите его. Если в строке несколько слов одинаковой максимальной длины, выведите первое из них. Решение оформите в виде функции LongestWord (S), возвращающей значение типа str.

| Ввод | Вывод |
|---|--------|
| In a hole in the ground there lived a hobbit. | ground |

P: Капитан Флинт

Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: "North 5", где первое слово — одно из "North", "South", "East",
" Loading [MathJax]/extensions/MathZoom.js шагов, необходимое пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось ОХ направлена на восток, ось ОУ – на север.

Программа получает на вход последовательность строк указанного вида, завершающуюся строкой со словом "Treasure!". Программа должна вывести два целых числа: координаты клада.

| Ввод | Вывод |
|---|-------|
| North 5 East 3 South 1 Treasure! | 3 4 |

Q: Слова с прописной буквы

Дана строка. Измените регистр символов в этой строке так, чтобы первая буква каждого слова была заглавной, а остальные буквы - строчными.

Решение оформите в виде функции Capitalize (S), возвращающей новую строку.

| Ввод | Вывод |
|---|---|
| In a hole in the ground there lived a hobbit. | In A Hole In The Ground There Lived A Hobbit. |

R: Значение выражения - 3

Дана строка, содержащая одно или более целых неотрицательных чисел, разделенных знаками "+" или "-". Вычислите значение этого выражения.

Решение оформите в виде функции Eval(S).

| Ввод | Вывод |
|---------|-------|
| 21+7-10 | 18 |

S: Поиск подстроки

Даны две строки, возможно, содержащие пробелы. Выведите слово YES, если первая строка является подстрокой второй строки или слово NO в противном случае.

Loading [MathJax]/extensions/MathZoom.js IsSubstring(Pattern, Source).

Нельзя использовать стандартные функции и методы поиска.

| Ввод | Вывод |
|---|-------|
| hole in the ground In a hole in the ground there lived a hobbit. | YES |
| hole on the ground In a hole in the ground there lived a hobbit. | NO |

Т: Самая частая буква

Дана строка, возможно, содержащая пробелы. Определите, какая буква латинского алфавита (или какие буквы) в этой строке встречается чаще всего. При решении этой задачи заглавные и строчные буквы считаются одинаковыми, а прочие символы, не являющиеся буквами, не учитываются.

Программа должна вывести в первой строке все буквы, которые встречаются чаще всего в исходной строке. Выводить буквы необходимо в заглавном написании, в алфавитном порядке, без пробелов. Во второй строке выведите единственное число - сколько раз в данной строке встречаются эти буквы.

При решении этой задачи нельзя пользоваться вложенными циклами. Входная строка должна обрабатываться за один проход.

| Ввод | Вывод |
|--|---------|
| - We all live in the Yellow Submarine! | EL 5 |

У: Палиндром - 2

Дана строка, возможно, содержащая пробелы. Определите, является ли эта строка палиндромом, при условии, что заглавные и строчные буквы не различаются, а все символы, не являющиеся буквами, должны быть пропущены. Выведите слово YES, если слово является палиндромом и словов NO, если не является.

Решение оформите в виде функции `IsPalindrome(S)`. Длина входной строки может быть до 100000 символов. При решении этой задачи нежелательно пользоваться дополнительными строками и списками, модифицировать исходную строку.

| | |
|--|--------------|
| Loading [MathJax]/extensions/MathZoom.js | |
| ВВОД | ВЫВОД |

| | |
|---------------------|-----|
| Was it a rat I saw? | YES |
| abca | NO |

V: IP-адрес

В сети интернет каждому компьютеру присваивается четырехбайтовый код, который принято записывать в виде четырех чисел, каждое из которых может принимать значения от 0 до 255, разделенных точками. Вот примеры правильных IP-адресов:

127.0.0.0
192.168.0.1
255.0.255.255

Программа получает на вход строку из произвольных символов. Если эта строка является корректной записью IP-адреса, выведите YES, иначе выведите NO.

| Ввод | Вывод |
|-----------|-------|
| 127.0.0.1 | YES |

W: Детская задача

Хорошо известна задача-шутка, в которой требуется продолжить числовой ряд:

1
11
21
1211
111221
312211
13112221
1113213211
...

Напишите программу, которая по данному числу $k \leq 25$ выводит k -е число в этом ряду.

| Ввод | Вывод |
|------|-------|
| 4 | 1211 |



Loading [MathJax]/extensions/MathZoom.js

Дана текстовая строка, содержащая буквы латинского алфавита, пробелы, запятые и точки. Отформатируйте этот текст по следующим правилам:

1. В начале и конце строки не должно быть пробелов.
2. Все слова разделяются ровно одним пробелом.
3. Точки и запятые пишутся слитно с предыдущим словом, после знака препинания ставится пробел.

Выведите полученную строку.

| Ввод | Вывод |
|-----------------|---------------|
| Hello , world . | Hello, world. |

Y: Календарь

Напечатайте календарь на месяц. Месяц может состоять из 28, 29, 30, 31 дня. Календарь на месяц состоит из 4, 5 или 6 столбцов, заполненных числами (датами). Ширина каждого столбца - два символа, между столбцами промежутков в один символ. Однозначные числа дополняются пробелом слева. Промежуток между столбцами - один пробел. В каждом столбце должно быть записано хотя бы одно число.

Программа получает на вход два числа - количество дней в месяце (от 28 до 31) и день недели, на которое приходится первое число месяца (от 1 до 7) и должна вывести календарь на указанный месяц.

| Ввод | Вывод |
|------|---|
| 31 3 | 6 13 20 27 7 14 21 28 1 8 15 22 29 2 9 16 23 30 3 10 17 24 31 4 11 18 25 5 12 19 26 |

Z: Значение выражения - 4

Строка состоит из целых неотрицательных чисел, разделенных знаками операций "+", "-" и "*". Вычислите значение этого выражения выполняя действия по правилам арифметики.

Тесты к этой задаче закрыты

Loading [MathJax]/extensions/MathZoom.js

| Ввод | Вывод |
|----------|-------|
| $10-2*3$ | 4 |