

Оглавление

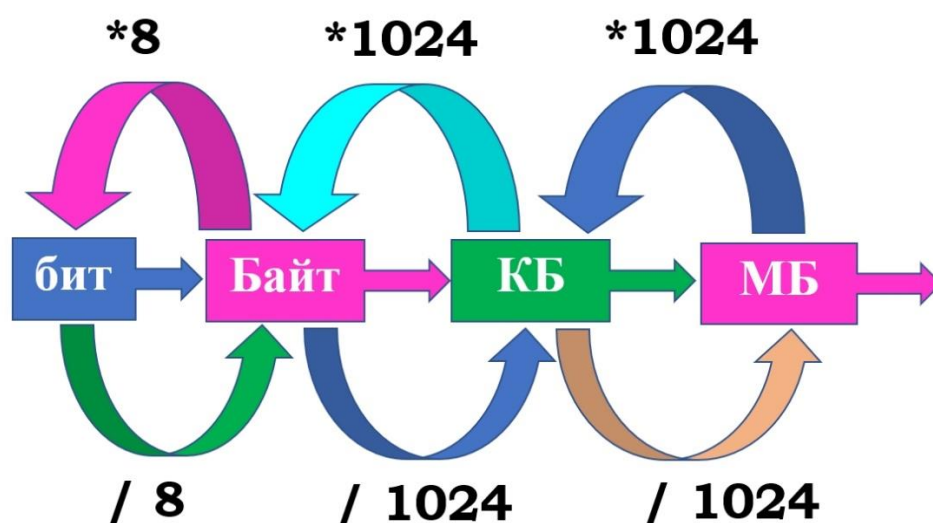
Основные формулы.....	2
Необходимые модули языка Python.....	3
Язык программирования Python.....	4
Алгебра логики.....	5
Рекурсия.....	5
Системы счисления.....	6
Если нужен АЛФАВИТ:.....	7
Списки.....	8
Основные виды алгоритмов.....	9
Некоторые команды Python:.....	11
Маска и адрес сети.....	12
Комбинаторика.....	13
Делители и Маска числа.....	14
Чтение чисел из файла:.....	14
Команды в языке Python.....	15
Электронные таблицы.....	17

Основные формулы

Графика:	$N = 2^i$ N – палитра цветов i – глубина цвета
	$F = i * m * n$ $m * n$ – разрешение
Звук:	$F = \Gamma_{\text{ц}} * \text{бит} * t_{\text{с}} * \text{канал}$ 1 моно 2 стерео 4 квадро
Текст	$N = 2^i$ N – мощность алфавита i – вес символа
	$F = i * m$ m – длина текста (количество символов)
Сеть:	$F = v * t_{\text{с}}$
МБ → КБ	МБ * 1024
КБ → бит	КБ * 1024 * 8
бит → КБ	бит / 2^{13}

n	0	1	2	3	4	5	6	7	8
2^n	1	2	4	8	16	32	64	128	256

n	9	10	11	12	13	14	15	16	17
2^n	512	1024	2048	4096	8192	16384	32768	65536	131072



Необходимые модули языка Python

from turtle import* - исполнитель черепаха
from itertools import* - для комбинаторики (размещения, перестановки)
from functools import* - @lru_cache()
from math import* - математический модуль (sqrt, factorial())
from statistics import* - (*среднее арифметическое, среднее геометрическое, среднее гармоническое, медиана последовательности, список наиболее часто встречающихся значений элементов*)
from collections import* - для частотного анализа (текста, списка...)
from re import* - регулярные выражения
from sys import* - изменять глубину рекурсии
from array import* - массивы
from fnmatch import* - поиск текста по маске
from ipaddress import* - IP-адрес и маска сети

a=[] – список

b = {} – словарь

c = set() - множество

Язык программирования Python

Операции

+ — сложение чисел/строк

- — вычитание

* — умножение числа или дублирование строки

** — возведение в степень

pow(x,y[,z]) — x^y по модулю (если модуль задан) = $x**y \% z$

/ — деление

// — частное от деления нацело — аналог div

% — остаток от деления нацело — аналог mod

divmod(x, y) — Возвращает пару значений (x//y, x%y).

Invariant: div*y + mod == x.

abs() — Модуль числа

!= — проверяет неравенство значений двух объектов.

== — проверяет равенство значений двух объектов.

is — проверяет идентичность самих объектов, включая тип объекта (аналог `===` ?).

= — присвоение.

:= — моржовый (walrus) оператор, появившийся в Python 3.8, дает возможность решить сразу две задачи: присвоить значение переменной и вернуть это значение.

Округление

floor(x) — Округляет число вниз («пол»), при этом floor(1.5) == 1, floor(-1.5) == -2

ceil(x) — Округляет число вверх («потолок»), при этом ceil(1.5) == 2, ceil(-1.5) == -1

Корни, логарифмы

sqrt(x) — Квадратный корень. Использование: sqrt(x)

log(x) — *Натуральный* логарифм.

log(x, b) — Логарифм по основанию b.

e — Основание натуральных логарифмов $e = 2,71828...$

Тригонометрия

sin(x) — Синус угла, задаваемого в радианах

cos(x) — Косинус угла, задаваемого в радианах

tan(x) — Тангенс угла, задаваемого в радианах

asin(x) — Арксинус, возвращает значение в радианах

acos(x) — Арккосинус, возвращает значение в радианах

atan(x) — Арктангенс, возвращает значение в радианах

atan2(y, x) — Полярный угол (в радианах) точки с координатами (x, y).

degrees(x) — Преобразует угол, заданный в радианах, в градусы.

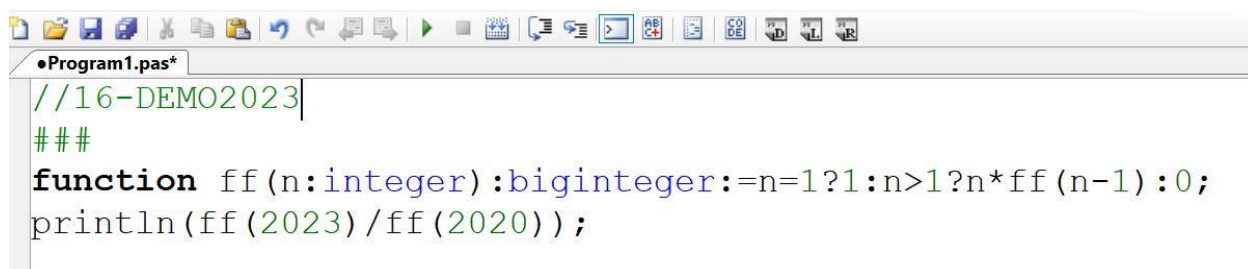
radians(x) — Преобразует угол, заданный в градусах, в радианы.

pi — Константа $\pi = 3.1415...$

Алгебра логики

Обозначение	Название	ФОРМУЛА	PYTHON	PascalABC	C++
$\bar{A}$	Инверсия	НЕ	not A	not A	!A
$A \wedge B$	Конъюнкция	И	A and B	A and B	A && B
$A \vee B$	Дизъюнкция	ИЛИ	A or B	A or B	A B
$A \rightarrow B$	Импликация	$\bar{A} \vee B$	A<=B	A <= B	A <= B !A B
$A \equiv B$	Эквиваленция	$A \wedge B \vee \bar{A} \wedge \bar{B}$	A==B	A = B	A == B
$A \oplus B$	Исключ ИЛИ	$A \wedge \bar{B} \vee \bar{A} \wedge B$	A ^ B	A xor B	A xor B

Рекурсия



```

//16-DEMO2023
###
function ff(n:integer):biginteger:=n=1?1:n>1?n*ff(n-1):0;
println(ff(2023)/ff(2020));

```

На языке Python для рекурсии:

```

from functools import*
from sys import*
setrecursionlimit(999999)
@lru_cache()

```

Системы счисления

```
type help , copyright ,
>>> bin(1234)
'0b10011010010'
>>> bin(1234)[2:]
'10011010010'
>>> oct(1234)[2:]
'2322'
>>> hex(1234)
'0x4d2'
>>> hex(1234)[2:].upper()
'4D2'
>>> format(1234,'b')
'10011010010'
>>> format(1234,'o')
'2322'
>>> format(1234,'x')
'4d2'
>>> format(1234,'x').upper()
'4D2'
```

```
>>> f'{1234:b}'
'10011010010'
>>> f'{1234:o}'
'2322'
>>> f'{1234:x}'
'4d2'
>>> f'{1234:x}'.upper()
'4D2'
```

```
> int('110011',2)
51
> int('1234',5)
194
> int('1234',8)
668
> int('57fd4',16)
360404
> int('9fg7k5',23)
62323542
```

Функция перевода десятичного числа в систему $Q < 10$:

```
def ff(a,b):
    s = ""
    while a > 0:
        s = str(a%b)+ s
        a = a//b
    return s
```

ПЕРЕВОД ДЕСЯТИЧНОГО ЧИСЛА В ЛЮБУЮ СИСТЕМУ:

```
1 t='0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
2 def ff(a,b):
3     s=''
4     while a>0:
5         s=t[a%b]+s
6         a=a//b
7     return s
8 print(ff(1234,22))
9 print(ff(1234,16))
0 print(ff(1234,23))
1 print(ff(1234,7))
2
```

```
= RESTART
2C2
4D2
27F
3412
```

Если нужен АЛФАВИТ:

```
>>> ''.join(sorted('qwertyuiopasdfghjklzxcvbnm')).upper()
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
>>> ''.join(sorted('йцукенгшщзхъфывапролджэячсмитьбю')).upper()
'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
```

Списки

```
>>> a=[12,33,4,5,9,11]
>>> sum(a)
74
>>> len(a)
6
>>> min(a)
4
>>> max(a)
33
>>> sum(a[2:7])
29
```

Команды для работы со списками

```
>>> from statistics import*
>>> a=[12,34,55]
>>> mean(a)
33.666666666666664
```

Среднее арифметическое списка

`a=[]`

`a.append(x)` – добавление значений в список

`b=set()`

`b.add(x)` – добавление значений в множество

множество убирает дубликаты

`if len(set(s))==len(s):` все цифры числа или символы строки различны

Основные виды алгоритмов

1) Ветвление

Python

if условие:

else:

2) Цикл с параметром Python

for n in range(2, 10):

... for x in range(2, n):

3) Цикл с предусловием Python

while условие:

while a < n:

Сумма цифр строки:

Python:

```
print(sum(map(int,s)))
```

Функция на проверку простоты числа:

```
def ff(n):
```

```
    for i in range(2,int(n**0.5)+1):
```

```
        if n%i==0: return False
```

```
    return True
```

```
if ff(x): k+=1
```

Конструкция match-case в Python**Пример с if-elif-else**

```
day = "Monday"
if day == "Sunday" {
    print("ВЫСПАТЬСЯ!")
}
else if day == "Monday" {
    print("На работу")
}
else if day == "Tuesday" {
    print("Боулинг!")
}
else if day == "Wednesday" {
    print("Общение")
}
else if day == "Thursday" {
    print("Выставка")
}
else if day == "Friday" {
    print("РЫНОК!")
}
else if day == "Saturday" {
    print("Пробежка!")
}
```

Заменим if-else на match-case

```
day = input("День=")
match day:
    case "Sunday" : print("ВЫСПАТЬСЯ!")
    case "Monday" : print("На работу ")
    case "Tuesday" : print("Боулинг!")
    case "Wednesday": print("Общение ")
    case "Thursday" : print("Выставка ")
    case "Friday" : print("РЫНОК!")
    case "Saturday" : print("Пробежка!")
```

Некоторые команды Python:

```
a,b=divmod(1234,38)
```

```
a = 32
```

```
b = 18
```

```
>>> divmod(124,16)
(7, 12)
```

```
>>> from math import*
>>> factorial(7)
5040
>>> gcd(170,40)
10
>>> gcd(1234,88)
2
>>> cbirt(512)
8.0
```

gcd(a,b) - наибольший общий делитель

cbirt(x) – кубический корень ($\sqrt[3]{X}$)

```
>>> from collections import*
>>> s='1234577654443328977775444332221111187433347777'
>>> Counter(s)
Counter({'7': 11, '4': 9, '3': 8, '1': 6, '2': 5, '5': 3, '8': 2,
'6': 1, '9': 1})
>>> Counter(s).most_common(1)
[('7', 11)]
>>> Counter(s).most_common(4)
[('7', 11), ('4', 9), ('3', 8), ('1', 6)]
```

Маска и адрес сети

IP адрес:	165.198.72.135
Маска:	255.255.192.0
IP адрес сети:	165.198.X.0

$$72_{10} = 01001000_2$$

$$192_{10} = 11000000_2$$

IP адрес сети: 165.198.64.0

```
from ipaddress import*  
net = ip_network("145.168.35.214/255.255.248.0", 0)  
print(net.network_address)
```

Комбинаторика

```
>>> from itertools import *
>>> ''.join(list(product('АКЛОШ', repeat=5)) [345])
'АЛОША'
>>> ''.join(list(product('ТИМАШЕВСК', repeat=4)) [1234])
'ИВМИ'
```

```
from itertools import *
```

```
for i in permutations('abc'):
    print(i, end=' ')
```

или

```
print( [''.join(x) for x in permutations('abc')])
```

```
for i in product('abc', repeat=2):
    print(i, end=' ')
```

```
for i in combinations('abcd', 2):
    print(i, end=' ')
```

```
for i in combinations_with_replacement('abcd', 2):
    print(i, end=' ')
```

Делители и Маска числа

```
from fnmatch import*  
if fnmatch(str(x), '12*5??*8'): print(x)
```

Чтение чисел из файла:

Pascal:

```
var a:=Readalltext('17.txt').ToIntegers;
```

Python:

```
A=[int(x) for x in open('17.txt')]
```

```
B = min(x for x in a if x%100==25)
```

```
C = max(x for x in a if x%2==0)
```

Команды в языке Python

```
>>> x=1234567
>>> x%100
67
>>> x%1000
567
>>> x//1000
1234
>>> x//100%10
5
>>> x//1000%100
34
```

% - остаток от деления

// - деление нацело или `int(x/100)`

```
>>> x=1234567
>>> x//1000
1234
>>> int(x/1000)
1234
```

```
>>> x=12345
>>> y=7658
>>> x+y
20003
>>> str(x)+str(y)
'123457658'
>>> str(y)+str(x)
'765812345'
>>>
```

str – перевод числа в текст

```
s=s.replace('A','1').replace('B','2').replace('C','3').replace('D','4')
for c in 'ABCDEFGH': s=s.replace(c,'*')
```

два способа ЗАМЕНЫ символов в строке

```
>>> s='1234567'
>>> s[-1]
'7'
>>> s[-2]
'6'
>>> s[:-3]
'1234'
>>> s[-4:]
'4567'
>>> s[::-1]
'7654321'
```

[] – срезы в строках

Строка наоборот

```
>>> s='348765909217'
>>> s[3]+s[4]
'76'
>>> s[0]+s[10]
'31'
>>> s[:5]+s[-2:]
'3487617'
```

```
>>> a=[1,2,3,4,5,6,7,8,9]
>>> a[:5]
[1, 2, 3, 4, 5]
>>> a[::-1]
[9, 8, 7, 6, 5, 4, 3, 2, 1]
>>> a[:3]+a[-3:]
[1, 2, 3, 7, 8, 9]
```

Электронные таблицы

MS EXCEL	LibreOffice CALC
=СУММ(A4:H7)	=СУММ(A4:H7)
=СУММЕСЛИ(A4:H7;"<77")	=СУММЕСЛИ(A4:H7;"<77")
=СУММЕСЛИМН()	=СУММЕСЛИМН()
=МАКС(A4:H7)	=МАКС(A4:H7)
=МИН(A4:H7)	=МИН(A4:H7)
=НАИМЕНЬШИЙ(A1:D1;1)	=НАИМЕНЬШИЙ(A1:D1;1)
=НАИМЕНЬШИЙ(A1:D1;2)	=НАИМЕНЬШИЙ(A1:D1;2)
=НАИБОЛЬШИЙ(A1:D1;1)	=НАИБОЛЬШИЙ(A1:D1;1)
=НАИБОЛЬШИЙ(A1:D1;2)	=НАИБОЛЬШИЙ(A1:D1;2)
=СРЗНАЧ(A1:D1)	=СРЗНАЧ(A1:D1)
=СРЗНАЧЕСЛИ(A1:D1;"<77")	=СРЗНАЧЕСЛИ(A1:D1;"<77")
=СЧЕТ (A1:D1)	=СЧЕТ (A1:D1)
=СЧЕТЕСЛИ(A1:D1;"1")	=СЧЕТЕСЛИ(A1:D1;"1")
=ЕСЛИ(условие;1;0)	=ЕСЛИ(условие;1;0)
=ЕСЛИ(И(условие; условие; условие);1;0)	=ЕСЛИ(И(условие; условие; условие);1;0)
=ОСТАТ(A1;2) – остаток от деления	=ОСТАТ(A1;2) – остаток от деления
=ОСНОВАНИЕ(A1;23)	=ОСНОВАНИЕ(A1;23)
=ОСНОВАНИЕ(A1;7) – перевод в системы счисления до 36	=ОСНОВАНИЕ(A1;7) – перевод в системы счисления до 36