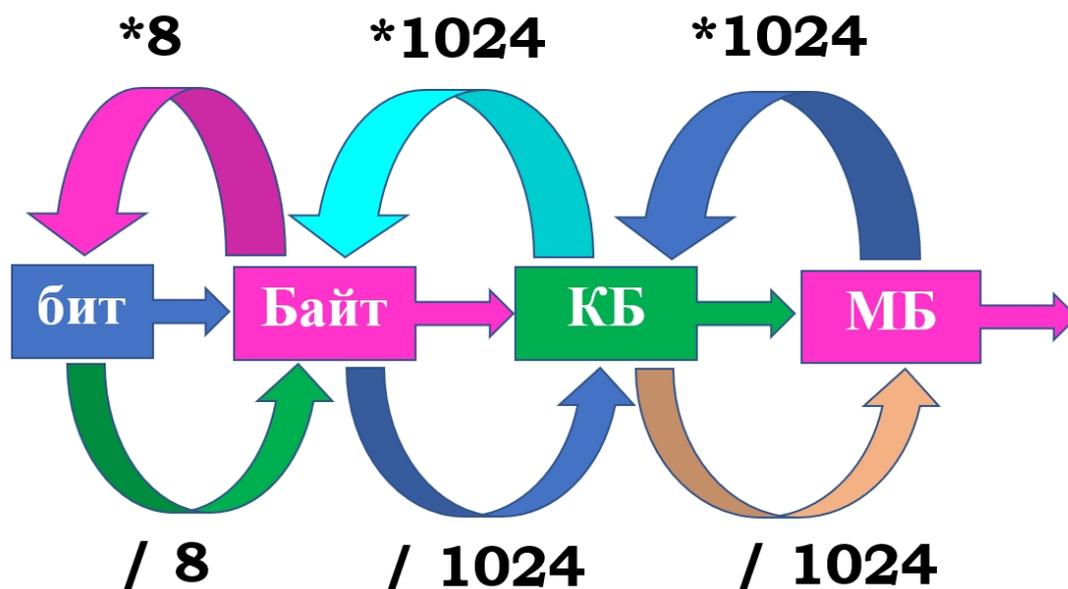


<b>Графика:</b>	$N = 2^i$ N – палитра цветов i – глубина цвета
	$F = i * m * n$ m*n – разрешение
<b>Звук:</b>	$F = \Gamma_{ц} * \text{бит} * t_c * \text{канал}$ 1 моно 2 стерео 4 квадро
<b>Текст</b>	$N = 2^i$ N – мощность алфавита i – вес символа
	$F = i * m$ m – длина текста (количество символов)
<b>Сеть: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

n	0	1	2	3	4	5	6	7	8
2 <sup>n</sup>	1	2	4	8	16	32	64	128	256

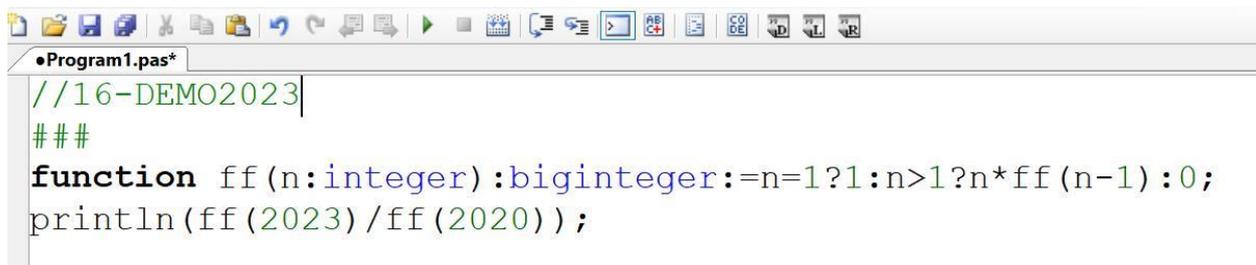
n	9	10	11	12	13	14	15	16	17
2 <sup>n</sup>	512	1024	2048	4096	8192	16384	32768	65536	131072



## 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 int, 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'
```

**В языке PascalABC.NET встроен перевод десятичных чисел в любую систему счисления:**

```

Глобальные имена
1  ### uses school;
2  println(tobase(123456, 5), tobase(123456, 21));

```

Окно вывода

12422311 D6JI

**Если нужен АЛФАВИТ:**

```

>>> ''.join(sorted('qwertyuiopasdfghjklzxcvbnm')).upper()
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
>>> ''.join(sorted('йцукентгшщзхъфывапроджэячсмитьбю')).upper()
'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'

```

```

>>> 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

## Ветвление

### **PascalABC.NET**

if <условие> then <оператор 1> else <оператор 2>;

### **Python**

if условие:

-----

else:

## Цикл с параметром Python

for n in range(2, 10):

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

## Цикл с параметром Pascal

for i:=12 to 67 do begin

end;

## Цикл с предусловием Python

while условие:

while a < n:

## Цикл с предусловием Pascal

While условие do

While a<n do begin

end;

**Сумма цифр строки:***PascalABC.Net*

```
println(s.Sum(d->strtoint(d)));
```

*Python:*

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

**Некоторые команды Python:**

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

```
a = 32
```

```
b = 18
```

*from math import\**

```
>>> 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='123457765444332897777544433222111187433347777'
>>> 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)
```

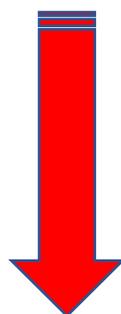
```
1  ### uses school;
2  var ip:=new CalcIP('148.168.99.45', '255.255.240.0');
3  println(ip);
4  println;
5  var a := new Addr32('192.168.0.1');
6  println(a);
7
```

---

Окно вывода

IP адрес (Address)	148.168.99.45	10010100.10101000.0110	0011.00101101
Префикс маски подсети (Bitmask)	/20		
Маска подсети (Netmask)	255.255.240.0	11111111.11111111.1111	0000.00000000
Маска хостов (Wildcard)	0.0.15.255	00000000.00000000.0000	1111.11111111
IP адрес сети (Network)	148.168.96.0	10010100.10101000.0110	0000.00000000
Широковещательный адрес (Broadcast)	148.168.111.255	10010100.10101000.0110	1111.11111111
Доступно адресов для хостов (Hosts)	4094		
Адрес первого хоста (Hostmin)	148.168.96.1	10010100.10101000.0110	0000.00000001
Адрес последнего хоста (Hostmax)	148.168.111.254	10010100.10101000.0110	1111.11111110

192.168.0.1, 11000000.10101000.00000000.00000001



## Комбинаторика

```

1  ###
2  '123'.cartesian(4).Prln;
3  Prln;
4  'ABCD'.permutations.Prln;
5  Prln;
6  '123789'.combinations(4).Prln;
7  Prln;

```

Окно вывода

```

1111 1112 1113 1121 1122 1123 1131 1132 1133 1211 1212 1213 1221 1222 1223 1231 1232 1233 1311
2113 2121 2122 2123 2131 2132 2133 2211 2212 2213 2221 2222 2223 2231 2232 2233 2311 2312 2313
3122 3123 3131 3132 3133 3211 3212 3213 3221 3222 3223 3231 3232 3233 3311 3312 3313 3321 3322

ABCD ABDC ACBD ACDB ADBC ADCB BACD BADC BCAD BCDA BDAC BDCA CABD CADB CBAD CBDA CDAB CDBA DABC

1237 1238 1239 1278 1279 1289 1378 1379 1389 1789 2378 2379 2389 2789 3789

```

```

>>> 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=' ')
```

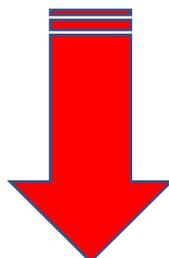
ИЛИ

```
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

*Получить список делителей числа:*

```
### uses school;  
var d:=x.Divisors;  
var f:=x.DivisorsCount;  
PrimeDivisorsCount возвращает количество простых  
делителей 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')]
```

```
>>> 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  $\text{int}(x/100)$

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

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

str – перевод числа в  
текст

```
>>> 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]
```

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)