

Преобразуване в бройни системи (БС)

- Да преобразуваме число от една в друга бройна система означава най-просто да запишем числото с други цифри.
- Числата в двоична БС записваме с долен десен индекс 2, ограден в скоби, напр. $10_{(2)}$. Аналогично за всички БС, освен десетична.

От десетична в двоична:

$244 \rightarrow 11110100_{(2)}$

$244 : 2 = 122$, ост. **0**

$122 : 2 = 61$, ост. **0**

$61 : 2 = 30$, ост. **1**

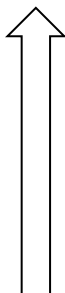
$30 : 2 = 15$, ост. **0**

$15 : 2 = 7$, ост. **1**

$7 : 2 = 3$, ост. **1**

$3 : 2 = 1$, ост. **1**

$1 : 2 = 0$, ост. **1**



От двоична в десетична:

$11110100_{(2)} \rightarrow 244$

$0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 + 1 \cdot 2^7 = 4 + 16 + 32 + 128 = 244$

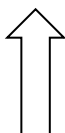
От десетична в осмична:

$244 \rightarrow 364_{(8)}$

$244 : 8 = 30$, ост. **4**

$30 : 8 = 3$, ост. **6**

$3 : 8 = 0$, ост. **3**



От осмична в десетична:

$364_{(8)} \rightarrow 244$

$4 \cdot 8^0 + 6 \cdot 8^1 + 3 \cdot 8^2 = 4 + 48 + 192 = 244$

От шестнадесетична в десетична:

$*a = 10, b = 11, c = 12, d = 13, e = 14, d = 15*$

$11a_{(16)} \rightarrow 282$

$10 \cdot 16^0 + 1 \cdot 16^1 + 1 \cdot 16^2 = 10 + 16 + 256 = 282$

Степени на двойката

$2^0 = 1$	$2^4 = 16$	$2^8 = 256$
$2^1 = 2$	$2^5 = 32$	$2^9 = 512$
$2^2 = 4$	$2^6 = 64$	$2^{10} = 1024$
$2^3 = 8$	$2^7 = 128$	$2^{11} = 2048$

Операции с двоични числа

Събиране:

$1110_{(2)}$
 $+ 0111_{(2)}$

 $10101_{(2)}$

A	B	СУМА	ПРЕНОС
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Изваждане:

$1110_{(2)}$
 $- 0111_{(2)}$

 $0111_{(2)}$

A	B	РАЗЛИКА	ЗАЕМ
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Умножение:

$1110_{(2)}$
 $* 0111_{(2)}$

 $0110\ 0010_{(2)}$

A	B	A x B
0	0	0
0	1	0
1	0	0
1	1	1

Деление:

$01000110_{(2)}$
 $/ \quad 111_{(2)}$

 $1010_{(2)}$

A	B	A/B
0	1	0
1	1	1

Побитови (bitwise) операции

& (bitwise AND) –

връща 1, ако и двата бита са 1.

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

/ (bitwise OR) –

връща 1, ако поне 1 от битовете е 1.

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

^ (bitwise XOR) –

връща 1, ако битовете са различни.

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

<< (left shift) - взима 2 числа, първото е това, на което ще се местят (shift) битовете, второто посочва с колко позиции наляво.

>> (right shift) - взима 2 числа, първото е това, на което ще се местят (shift) битовете, второто посочва с колко позиции надясно.

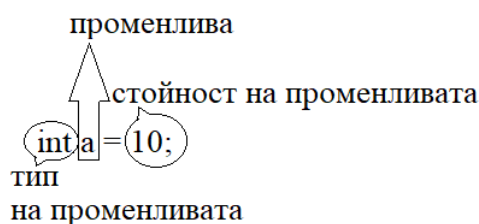
~ (bitwise NOT) - взима число и му обръща битовете. Всички 0 стават 1, а всички 1 стават 0. За целочислено число N, операцията ни дава - (N + 1).

Мерни единици за памет

Бит	bit	b	1/0
Байт	byte	B	1B = 8b
Килобайт	kilobyte	KB	1KB = 2 ¹⁰ B
Мегабайт	megabyte	MB	1MB = 2 ²⁰ B
Гигабайт	gigabyte	GB	1GB = 2 ³⁰ B
Терабайт	terabyte	TB	1TB = 2 ⁴⁰ B

Променливи и типове

Променлива – еквивалента на понятието променлива от математиката; можете да си мислите за x, y.



Горният ред означава, че в програмата ни вече съществува променлива, която се казва „a“ и нейната стойност е равна на 10.

Тип на променлива

Всяка променлива си има тип, който се определя от това какво искаме да се съдържа в променливата. Например, ако искаме да се съдържат цели числа, променливата трябва да е от тип int.

Някои типове:

int	цели числа
double	дробни числа
float	дробни числа
bool	двоични (булеви) числа
char	символи
unsigned int	неотрицателни числа
long long int	големи цели числа

Големина на променлива

Всяка променлива има големина в паметта на компютъра. Големината се измерва в байтове (B) и зависи от типа на променливата.

Големина:

int	4B
double	8B
float	4B
bool	1B
char	1B
unsigned int	4B
long long int	8B

Обхват на типовете

Всеки тип има обхват на стойностите, които може да приема. Когато зададем типа на една променлива, задаваме интервал от стойности, които променливата може да приема.

Обхват:

int	[-2147483648, 2147483647]
double	+/- 1.7e +/- 308 (~15 digits)
float	+/- 3.4e +/- 38 (~7 digits)
bool	true (1) or false (0)
char	[-127, 127] или [0, 255]
unsigned int	[0, 4294967295]
long long int	[-(2 ⁶³), (2 ⁶³)-1]

Създаване и въвеждане на променлива

```
int age;  
std::cin >> age;
```

Извеждане на променлива

```
std::cout << age;
```

Вградени функции и операции над числа

Вградените функции са функции, за които има запазена дума в езика.

Нека за примера имаме създадени 3 променливи int a = 5, int b = 3, double c = 7.2, int d, double e, int f = -3;

Събиране	+	d=a+b; //d == 8
Изваждане	-	d=a-b; //d == 1
Умножение	*	d=a*b; //d == 8
Целочислено деление	/	d=a/b; //d == 8
Остатък при деление	%	d=a%b; //d == 8
Деление	/, като поне едно от двете числа, които делим, е double	d=c/b; //d == 8
Степенуване	pow(a, b)	d = pow(a,b); //d == 8
[x]	floor(x)	d = floor(c);
[x]	ceil(x)	d = ceil(c); //d == 8
Закръгляне на число	round(x)	d = round(c); //d == 7;
√x	sqrt(x)	e = sqrt(b); //e == 1.732

$\log_a b$	$\log(b)/\log(a)$	$e = \log(a)/\log(b)$ // $e == 1.464$
e^x	$\exp(x)$	$e = \exp(b);$ // $e == e ^ 3;$
Тригонометрични	$\sin(x), \cos(x)$ и др., където x е в радиани	$\sin(0) == 0;$
Модул	$\text{abs}(x)$	$d = \text{abs}(f);$ // $d == 3;$

Операции за сравнение

Резултатът от тях е 1 (true) или 0 (false).

Равно	<code>==</code>
По-малко	<code><</code>
По-голямо	<code>></code>
По-малко или равно	<code><=</code>
По-голямо или равно	<code>>=</code>
Различно	<code>!=</code>

Логически операции

Логическите операции са операции, които се прилагат върху булеви променливи.

Конюнкция (и) - `&&`

Пр.:

`bool a = true;`

`bool b = false;`

`bool c = a && b;`

// `c == false;`

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

Дизюнкция (или) - `||`

Пр.:

`bool a = true;`

`bool b = false;`

`bool c = a || b;`

// `c == true;`

A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Негация (не) - `!`

Пр.:

`bool a = true;`

`bool c = !a;`

// `c == false;`

A	!A
0	1
1	0

Коментар в C++: с две наклонени черти пред

това, което искаме да закоментираме: `//` или

ограждаме секцията, която искаме да

закомнтираме, с `/*...*/`.

ASCII-таблица

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	