

Приоритет операций, хранение переменных

Семестр 1

Операнды, операции, операторы

- **Операнд** – переменные, константы и т.д.
 - *x, 12*
- **Операция** – символ/группа символов.
 - Арифметические -> бинарные
*+ - * / %*
 - Логические -> унарные
&& || ! > >= < <= == !=
 - Битовые
<< >> & ^ | ~
- **Оператор** – наименьшая исполняемая единица кода.
 - x = y;* <- оператор выражения
 - int d;* <- оператор объявления
 - {int x,y; x=5; y=30%x;}* <- составной оператор
 - for (int i=0; i<n; i++);* <- оператор цикла

Битовые операции

a = 0000 1111

b = 1000 1000

~a = 1111 0000 <- обращение

a << 1 = 0001 1110 <- сдвиг влево

a >> 1 = 0000 0111 <- сдвиг вправо

a & b = 0000 1000 <- и

a | b = 1000 1111 <- или

a ^ b = 1000 0111 <- исключающее или

Условная операция

Тернарная операция (три аргумента).

$y = x ? a : b;$ \leftarrow если $x \neq 0$, $y = a$;
если $x == 0$, $y = b$.

$y = (a > b) ? a : b;$ \leftarrow если $a > b$, $y = a$;
если $a \leq b$, $y = b$.

Приоритет операций

Уровень приор.	Символы	Описание	Порядок выполнения
1	++	Префиксный инкремент	Слева направо
	--	Префиксный декремент	
	()	Вызов функции и подвыражение	
	[]	Выделение элемента массива	
	->	Указатель структуры	
	.	Член структуры	
2	!	Логическое отрицание	Справа налево
	~	Поразрядное логическое НЕ (дополнение до 1)	
	-	Унарный минус (изменение знака)	
	+	Унарный плюс	
	(type)	Преобразование к типу	
	*	Обращение к памяти по значению указателя	
	&	Определение адреса переменной	
	sizeof	Определение размера в байтах	
3	*	Умножение	Слева направо
	/	Деление	
	%	Остаток от деления	
4	+	Сложение	Слева направо
	-	Вычитание	
5	<<	Поразрядный сдвиг влево	Слева направо
	>>	Поразрядный сдвиг вправо	
6	<	Меньше	Слева направо
	<=	Меньше или равно	
	>	Больше	
	>=	Больше или равно	
7	=	Равно	Слева направо
	!=	Не равно	
8	&	Поразрядное логическое И	Слева направо
9	^	Поразрядное исключающее ИЛИ (XOR)	Слева направо
10		Поразрядное логическое ИЛИ	Слева направо
11	&&	Логическое И	Слева направо
12		Логическое ИЛИ	Слева направо
13	?:	Операция условия	Слева направо
14	=	Присваивание	Слева направо
	+=	Составное сложение	
	-=	Составное вычитание	
	*=	Составное умножение	
	/=	Составное деление	
	%=	Составное определение остатка от деления	
	<<=	Составной поразрядный сдвиг влево	
	>>=	Составной поразрядный сдвиг вправо	
	&=	Составное поразрядное логическое И	
	^=	Составное поразрядное исключающее ИЛИ	
	=	Составное поразрядное логическое ИЛИ	
15	,	Операция «запятая»	Слева направо
	++	Постфиксный инкремент	
	--	Постфиксный декремент	

Приоритет операций

```
float a, b=2, c=1;  
a = b + c++;  
printf("%d %d %d",a,b,c);  
a = ++d + b;  
printf("%d %d %d",a,d,b);
```

```
if (c == 2 && b < 6)  
    printf("OK %d", c);  
else  
    printf("not quite OK %d", c);
```

```
if (c = 2 && b < 6)  
    printf("OK %d", c);  
else  
    printf("not quite OK %d", c);
```

Переменные

- Переменная - именованная область памяти, в которой хранится значение
- Размер букв имеет значение
- Имена переменных (идентификаторы)
 - a-z A-Z 0-9 _
 - не может начинаться с цифры
- что не идентификатор?

x12 X12 _x12

x_12 12X 12

Позиционные системы счисления

- Десятичная

$$615_{10} = 6 \cdot 10^2 + 1 \cdot 10^1 + 5 \cdot 10^0$$

10^3	10^2	10^1	10^0
0	6	1	5

- Двоичная

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$
$$8 + 4 + 1 = 13_{10}$$

$$2^3 = 8 \quad 2^2 = 4 \quad 2^1 = 2 \quad 2^0 = 1$$

1	1	0	1
---	---	---	---

2, 8 и 16 и двоично-десятичная система

DEC	BIN	OCT	HEX	BCD	
0	0000	0	0	0000	
1	0001	1	1	0001	
2	0010	2	2	0010	
3	0011	3	3	0011	
4	0100	4	4	0100	
5	0101	5	5	0101	
6	0110	6	6	0110	
7	0111	7	7	0111	
8	1000	10	8	1000	
9	1001	11	9	1001	
10	1010	12	A	0001	0000
11	1011	13	B	0001	0001
12	1100	14	C	0001	0010
13	1101	15	D	0001	0011
14	1110	16	E	0001	0100
15	1111	17	F	0001	0101

Двоично-десятичный код
(англ. binary-coded decimal)

36-ричная система
счисления (все буквы
+ цифры),
пример - серийный
номер программного
обеспечения

Часы для айтишников



Представление отрицательных чисел

$x + (-x) = 0$ в побитовом сложении

- Прямой код (знаковый бит)

3: 0011

-3: 1011

- Обратный код ($\sim x$)

3: 0011

-3: 1100

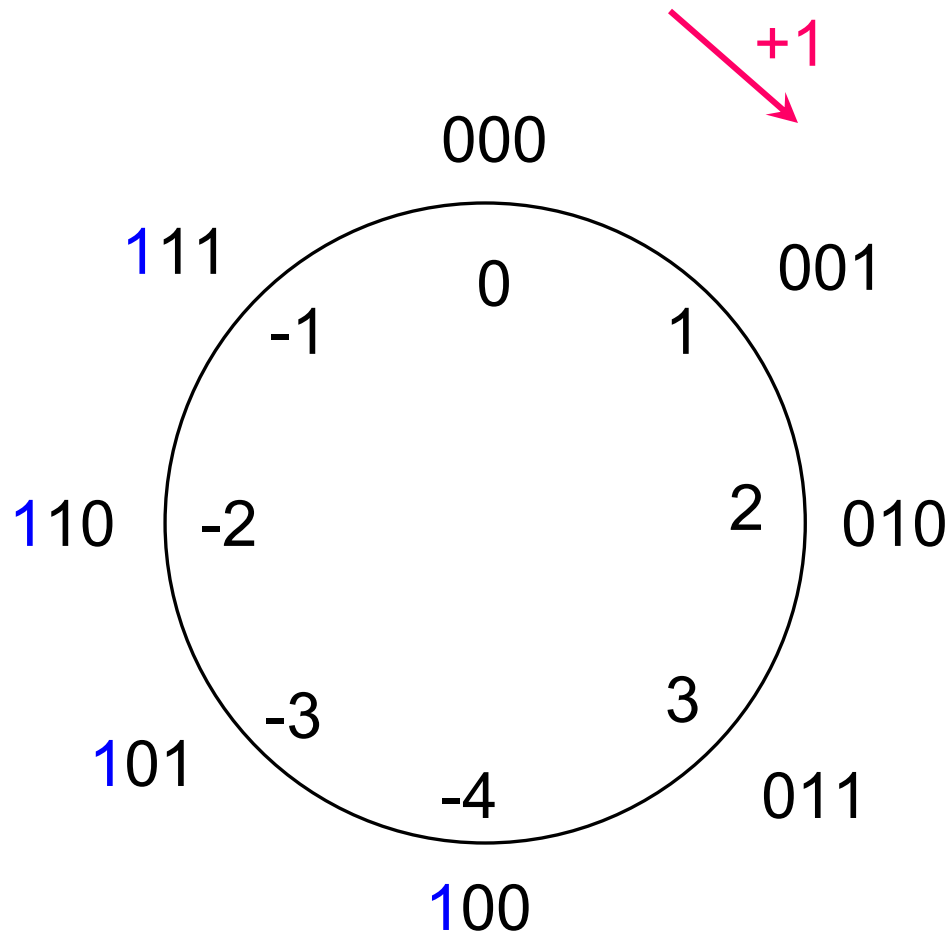
- Дополнительный код: $x + (-x) = 0$

3: 0011

-3: 1101 (обратный +1)

Σ : 10000

Диапазон знаковых чисел



- 3 битные числа
- $2^3 = 8$ чисел
- -4 ... 3

Типы данных

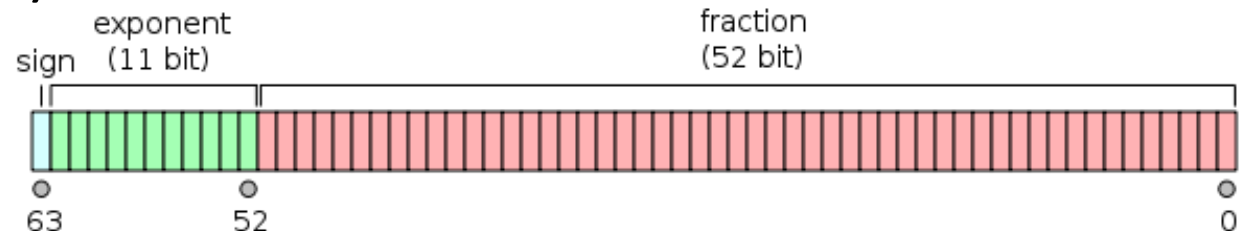
- целочисленные

- char (размер = 1 байту)
- short
- int
- long
- long long

unsigned long int x;

- дробные $(-1)^{\text{sign}} \cdot 2^{\text{exponent} - \text{exponent_bias}} \cdot 1.\text{mantissa}$

- float
- double



- модификаторы

- signed
- unsigned

Считать десятичное число

```
#include <stdio.h>
int main ( )
{
    int x, y;
    scanf ("%d", &x );
    scanf ("%d", &y );
    printf ("Sum is %d \n", x+y);

    return 0;
}
```

d - десятичное decimal
o - восьмеричное octal
h - шестнадцатеричное hex
u – беззнаковый (unsigned)

Задачи

- Преобразование десятичного числа в пятеричное.
- Преобразование десятичного числа в двоичное.
- Вывод `int` и `float` в компьютерном представлении (так, как хранятся в памяти).
- Преобразование двоичного числа в дополнительный код.