



Алгоритмы и Алгоритмические Языки

Семинар #3:

- Обратная связь по контексту #0;
- Базовые элементы языка;
- Битовые операции; Целочисленное деление;
- Сравнения и логические операции;
- Тернарная операция и операция “запятая”;



Обратная связь по контексту #0





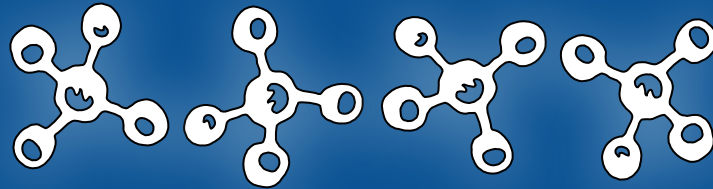
Контекст #0: сложности

Обратная связь – в системе тестирования.

Самые частые сложности в контексте #0:

- Система тестирования – не компилятор;
- Проверка вводов в программу в 00-2 и 00-3:
=> Возвращаемое значение `scanf()`;
- Переполнения при расчётах в 00-2 и 00-3:
=> Грамотный выбор типов, явное ограничение вводов;
- Непонятные сообщения об ошибках в 00-2 и 00-3;
=> Писать понятные сообщения об ошибках;
- Явная расчётная формула для 00-3:
<двойная арифметическая прогрессия>

Базовые элементы языка: операции, выражения и операторы



Операции (operators)

Операция (operator) – символ или набор символов, задающий некоторое вычисление.

Операция (operator) имеет:

- Аргументность и ассоциативность;
- Значение (результат операции);
- Побочный эффект (изменение состояния памяти и состояния процессора);

Аргументность уже известных операций:

- Унарные операции: - (знак “минус”), + (знак “плюс”)
- Бинарные операции: + (сложение), = (присваивание)
- Тернарная операция: `_ ? _ : _` (условная операция)

Операции (operators)

Ассоциативность уже известных операций:

- Левоассоциативные: - (вычитание)
$$x - y - z = (x - y) - z$$
- Правоассоциативные: композиция функций (математика)
$$(f \cdot g \cdot h)(x) = (f \cdot (g \cdot h))(x) = f(g(h(x)))$$
- Ассоциативные: + (сложение)
$$x + y + z = (x + y) + z = x + (y + z)$$

Побочные эффекты операций:

- Операция "=" меняет значение левого операнда на значение правого.

Операторы (operators)



Common operators						
assignment	increment decrement	arithmetic	logical	comparison	member access	other
<pre>a = b a += b a -= b a *= b a /= b a %= b a &= b a = b a ^= b a <<= b a >>= b</pre>	<pre>++a --a a++ a--</pre>	<pre>+a -a a + b a - b a * b a / b a % b ~a a & b a b a ^ b a << b a >> b</pre>	<pre>!a a && b a b</pre>	<pre>a == b a != b a < b a > b a <= b a >= b</pre>	<pre>a[b] *a &a a->b a.b</pre>	<pre>a(...) a, b (type) a a ? b : c sizeof _Alignof (since C11)</pre>

Нужны уже сейчас

Выражения (expressions)

Выражение (expression) – имя переменной, или имя функции, или литерал, задающий константу, или операция над другими выражениями.

Выражение имеет:

- Значение (результат вычисления выражения);
- Тип (int, float, ...);
- Побочный эффект;

Примеры выражений:

`x = y`

`777UL`

`(a - 2) * 4`

`x + y + 14`

`“I am expression”`

`(a < b)? a : b`

`printf(“word\n”)`

`index < 10`

`(long) val`

Операторы (statements)

Оператор (statement) – языковая конструкция, задающая набор действий для выполнения.

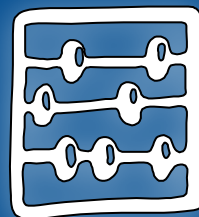
Операторы не имеют типа и значений, а имеют только побочные эффекты.

Примеры операторов:

- `expression;`
- `{ statement }`
- `if (expression) statement [else statement]`
- `while (expression) statement`
- `return expression;`
- ...



Битовые операции



Битовые операции

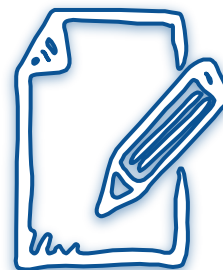
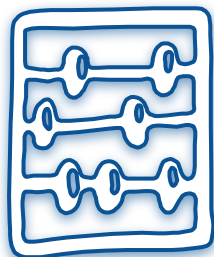


Битовые операции:

Синтаксис	Название
$\sim a$	Побитовое НЕ, NOT, инверсия
$a \mid b$	Побитовое ИЛИ, OR, дизъюнкция
$a \& b$	Побитовое И, AND, конъюнкция
$a \wedge b$	Побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ, XOR
$a \ll b$	Сдвиг a на b бит влево
$a \gg b$	Сдвиг a на b бит вправо

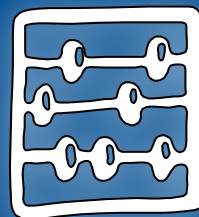
Битовые операции: задачи

1. Вычислить $x \cdot 2^y$;
2. Занулить 12 младших бит в 32-битном числе x ;
3. Поменять биты с 8 по 15 в в 32-битном числе x на противоположные;
4. Занулить k младшие бит беззнакового числа x ;
5. Изменить порядок следования байт в 32-битном числе x ;





Целочисленное деление



Целочисленное деление

Деление с остатком в математике: остаток – неотрицателен.

$$a \equiv r \pmod{b} \Leftrightarrow \exists q \in \mathbb{Z} : a = qb + r, 0 \leq r < b$$

(Остаток – неотрицательное число)

В языке Си: знак остатка совпадает со знаком делимого.

$$a \equiv r \pmod{b} \Leftrightarrow \exists q \in \mathbb{Z} : a = qb + r \\ -b < r < b, \text{sign } r = \text{sign } a$$

$$q = a / b;$$

$$r = a \% b;$$

Целочисленное деление: задача

1. Для заданных пар a и b произвести деление по модулю:

a	b	$q = a / b$	$r = a \% b$
25	8	???	???
25	-8	???	???
-25	8	???	???
-25	-8	???	???



Сравнения и логические операции



Сравнения и логические операции



Операции сравнения:

< (меньше)	> (больше)
<= (меньше или равно)	>= (больше или равно)
== (равно)	!= (не равно)

Логические операции:

&& (логическое И) ! (не) || (логическое ИЛИ)

Правило короткой логики:

- Если для && (или ||) левая часть выражения равна 0 (или 1), то правая часть выражения не вычисляется.

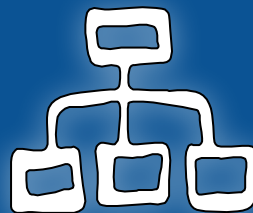


Сравнения: задачи

1. Записать выражение для условия:
“ x делится на y без остатка”;
2. Выписать логическое выражение, истинное **тогда и только тогда**, когда:
 - $x \in [-5, 19)$
 - $x \in (-\infty, 8] \cup [12, +\infty)$
 - $x \in (A, B] \cap [C, D)$



Тернарная операция и операция “запятая”



Тернарная операция и запятая

Оператор с тернарной операцией:

```
int x = (condition? compute_true : compute_false);
```

Эквивалентен оператору:

```
if (condition)
{
    x = compute_true;
}
else
{
    x = compute_false;
}
```

Тернарная операция и запятая

Оператор с операцией “запятая” служит для последовательного вычисления выражений.

Конструкция:

```
printf("Comma operator return value: %c\n", (print_a(), print_b()));
```

Эквивалентно:

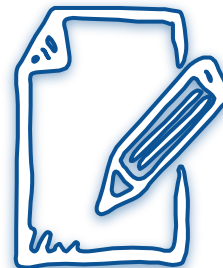
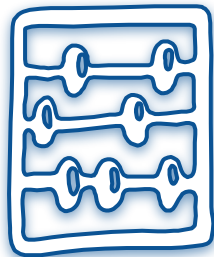
```
int print_ret = print_a();  
print_ret = print_b();  
printf("Comma operator return value: %c\n", (print_a(), print_b()));
```

См. пример `examples/03_side_effects`.

Тернарная операция: задачи

Выписать выражения для вычисления:

1. Модуля целого числа;
2. Максимума двух чисел.



Вопросы?

