

Вычисление выражения языка C

Составить рефал-программу, вводящую выражение, записанное на языке программирования C, и вычисляющую его значение. Пример такого выражения:

$x=2, y=3, z=x+=2*y-3*x?++y>4?(8-4)*x:25:x--\&\&y/3?x<<3:5$

Выражение состоит из целых констант, имён переменных, круглых скобок и может содержать следующие знаки операций:

- ✓ постфиксные и префиксные ++ и --;
- ✓ унарные и бинарные + и -;
- ✓ арифметические *, / и %;
- ✓ побитовые << и >>;
- ✓ логические !, &&, ||;
- ✓ операции отношения !=, ==, >, <, >=, <=;
- ✓ тернарная операция ? : ;
- ✓ операции присваивания =, +=, -=, *=, /=, %=>, <=>;
- ✓ операция запятая (,).

Приоритет и порядок вычисления этих операций соответствуют принятым в языке C [11] правилам – см. таблицу 1 (в ней операции располагаются по строкам в порядке убывания приоритета).

Синтаксис выражения языка C описывается следующими БНФ-правилами:

выражение ::= *выражение_присваивания* |

выражение , *выражение_присваивания*

выражение_присваивания ::= *условное_выражение* |

идентификатор операция_присваивания выражение_присваивания

операция_присваивания ::= = | += | -= | *= | /= |

%= | <=> | >>=

условное_выражение ::= *логическое_ИЛИ_выражение* |

логическое_ИЛИ_выражение ? выражение : условное_выражение

логическое_ИЛИ_выражение ::= *логическое_И_выражение* |

логическое_ИЛИ_выражение || *логическое_И_выражение*

...

слагаемое ::= *унарное_выражение* |

слагаемое *мульт_знак* *унарное_выражение*

мульт_знак ::= * | / | %

унарное_выражение ::= *постфиксное_выражение* |

унар_знак *унарное_выражение*

унар_знак ::= ! | ++ | -- | + | -

*постфиксное_выражение ::= первичное_выражение |
постфиксное_выражение1*

*постфиксное_выражение1 ::= идентификатор |
постфиксное_выражение1 постф_знак*

постф_знак ::= ++ | --

*первичное_выражение ::= идентификатор | целое_без_знака |
(выражение)*

При выполнении задания необходимо дописать недостающие БНФ-правила, и согласно получившемуся набору синтаксических правил составить рефал-программу, вычисляющую выражение языка С.

Таблица 1. Приоритет и порядок вычисления операций языка С

Приоритет операций (в порядке убывания)	порядок вычисления
постфиксные ++ и --	слева направо
! префиксные ++ и -- унарные + и -	справа налево
* / %	слева направо
бинарные + и -	слева направо
<< >>	слева направо
< <= > >=	слева направо
== !=	слева направо
&&	слева направо
	слева направо
? :	справа налево
= += -= *= /= %= <<= >>=	справа налево
,	слева направо