

## Первое задание по практикуму для самостоятельного программирования дома

Напишите программу-калькулятор выражений на языке C. После своего запуска программа должна выдать пользователю приглашение ко вводу (например, знак «>», «\$» и т.д.) и считать строку, введённую пользователем. Считанную строку программа должна интерпретировать как выражение на языке C, которое нужно вычислить вместе со всеми своими побочными эффектами. После вычисления выражения программа должна вывести на экран его значение и значения всех переменных, вывести приглашение ко вводу следующего выражения и повторять процедуру ввода выражений до тех пор, пока пользователь не введёт ключевое слово «exit».

В выражении допускаются:

1. Константы целого и вещественного типов, записанные по правилам языка C. Для простоты можно предполагать, что целые константы записаны в десятичной системе счисления и имеют тип `int`, а вещественные константы записаны в формате с фиксированной точкой, т.е. без экспоненты, и имеют тип `double`. Унарный минус перед константами может присутствовать, а может и отсутствовать.

Примеры:

(a) 123, -12, 0 — целые константы типа `int`.

(b) 1.23, -0.12, 1.0 — вещественные константы типа `double`.

2. Идентификаторы, записанные по правилам языка C, которые должны быть интерпретированы как имена переменных. Если идентификатор встречается в первый раз слева от операции присваивания, то объявляется переменная, имеющая значение и тип выражения, стоящего в правой части присваивания. Тип переменной далее в ходе работы программы не меняется. Ситуация, когда идентификатор в первый раз встречается справа от присваивания, является ошибочной, и в выражениях встретиться не может.
3. Знаки операций «+», «-», «/», «%», «=» и скобки «(», «)». Типы результатов и значения операций, а также их приоритет и ассоциативность должны быть такими же, как в языке C. Деление должно быть реализовано в семантике GCC.
4. Также вместо выражения может идти оператор `del <id>`, где `<id>` — идентификатор ранее объявленной переменной. Этот оператор удаляет переменную с именем `<id>`.

В примере работы программы ниже строки, начинающиеся с «>», отвечают выражениям, введённым пользователем. Остальные строки соответствуют выведенным программой значениям.

```
> a = 1 + 2 * (3 + 4 / -2)
Result: 3
a = 3
> b = a + 1
Result: 4
a = 3, b = 4
> b = (a + b) * 4
Result: 28
a = 3, b = 28
> c = 1.0 + a
Result: 4.0
a = 3, b = 28, c = 4.0
> c = a = (b = 1.0 * 3) + 2
Result: 5.0
a = 5, b = 3, c = 5.0
> del a
b = 3, c = 5.0
> b = c / 2
Result: 2
b = 2, c = 5.0
> c = b % 2
Result: 0.0
b = 2, c = 0.0
> a = 1.0 + 1
Result: 2.0
a = 2.0, b = 2, c = 0.0
> exit
```

Вывод программы может отличаться от приведённого выше. Можно считать, что все выражения корректны. В частности, не могут встретиться выражения вида `(a = 1) + (a = 2)`.

**Срок сдачи: 23:59, 27.10.2023.**