

Задание числа и системы счисления

Общие замечания

Требуется написать программу с полным контролем ошибок. По завершению работы терминал должен быть приведён в исходное состояние. Для работы с параметрами командной строки нужно использовать функции: *ParamCount* и *Paramstr*. Об этом можно прочитать здесь: <https://www.freepascal.org/docs-html/rtl/system/paramcount.html>.

Ко всем программам должны быть разработаны тесты демонстрирующие реакцию на корректный ввод, а так же на некорректный.

Пробельные символы (не путать с символом пробел) могут включать в себя пробелы, табуляции, перевод строки и возврат каретки.

Во всех программах может присутствовать не более чем 256-ричная система счисления. Для кодирования цифры в этой системе исчисления применяется алфавит:

`[0..9][a..z]`

Цифра записывается парой символов; 0 – записывается как '00', 10 – записывается как '0a', 16 записывается как 10, 17 – записывается как '11', и так далее. Запись 'ff' – соответствует числу 255.

1 Калькулятор чисел с плавающей точкой

В аргументах командной строки программе передаются следующие параметры:

- **epsilon** — такой что: $0 \leq \text{epsilon} \leq 1$, задаёт точность с которой нужно выводить дробную часть числа. Значение *epsilon* указывается в десятичной системе счисления.
- **num_base_1** [, **num_base_2** [, ... **num_base_n**]] — список систем счисления в которых необходимо вывести результат работы программы.

Программе на вход (в стандартном потоке ввода) передается текст в следующем формате. В тексте содержатся команды: строки с действительным числом **num** и двуместной операцией производимой с ним. Операции следующие: '+', '-', '*', '/'. Для операции подразумевается следующий формат операции:

```
result := result - num;
```

Здесь **result** — это скрытая переменная (регистр) которую хранит калькулятор для хранения результата всех выполненных операций. Собственно значение результата выводится в конце работы программы. Число указывается вместе со своей системой счисления. Указывается в формате:

`система_счисления:целая_часть.дробная`

Система счисления указывается как число от 2-х до 256 включительно записанное в основании 10.

В тексте могут встречаться комментарии, пустые строки: строки состоящие из произвольного числа пробельных символов в том числе ни одного, строки по сути состоящие только из комментариев. Комментарий — это последовательность символов начинающаяся с ';' и продолжающаяся до конца строки.

Каждая команда калькулятору указывается на своей строке, Формат команды следующий:

`операция система_счисления:целая_часть.дробная`

перед операцией может идти сколько угодно пробельных символов возможно ни одного, перед числом может идти сколько угодно пробельных символов как минимум один, после числа может идти сколько угодно пробельных символов и комментариев. Внутри числа цифры (пары символов) могут быть отделены друг от друга пробелами.

После успешного считывания очередной команды ввод команд заканчивается корректно: либо при встрече строки состоящей из слова *finish*, либо в случае закрытия входного потока (eof вернёт true). Хотя бы одна команда

должна быть прочитана успешно, только в этом случае калькулятор должен выдать ответ, иначе сообщение об ошибке и текущее значение переменной **result**, после чего завершиться.

Скрытая переменная калькулятора **result** должна иметь тип `double` и быть инициализирована нулём.

Результат должен выводиться для каждой из указанных в аргументах систем счисления следующим образом.

- На каждой строчке сперва указывается система счисления,
- затем начиная с 7-ой позиции в строке печатается само число-результат.
- Число печатается в соответствующей системе счисления с необходимой точностью для числа знаков после запятой.

1.1 Пример

Для запуска программы следующим образом (Среда Unix подобных операционных систем)

```
./calculator 0.1 2 10
```

и в потоке ввода

```
+ 70: 01 05 . 2d ; первое число
* 2: -0100.00    ; второе число
```

```
finish
```

; здесь уже не читаем

В потоке вывода будет:

```
2      -01 00 00 01 00 01 01 01 . 00
10     -01 05 01 . 00
```

2 Калькулятор рациональных чисел

Аргументы программы те же, что и в предыдущем случае. В отличие от предыдущего случая, здесь число указывается в другом формате:

система_счисления:целая_часть_числитель/натуральная_часть_знаменатель

Число должно храниться в паре переменных: `longint` для числителя и `longword` для знаменателя. Необходимо проверять, попадание в диапазон при считывании.

Формат команды следующий:

операция система_счисления:целая_часть_числитель/натуральная_часть_знаменатель

Скрытая переменная калькулятора **result** должна так же задаваться парой чисел и быть инициализирована рациональным нулём $\frac{0}{1}$.

После каждой операции число нужно приводить к каноническому виду (несократимой дроби).

Результат должен выводиться для каждой из указанных в аргументах систем счисления следующим образом: на каждой строчке сперва система счисления, затем с 7-ой позиции в строке само число результат как рациональное число, затем на новой строчке оно же в форме с плавающей точкой с необходимой точностью для числа знаков после запятой.

2.1 Пример

Для запуска программы следующим образом (Среда Unix подобных операционных систем)

```
./calculator 0.1 2 10
```

и в потоке ввода

```
+ 70: 01 05 / 2d ; первое число
- 2: -0101/01    ; второе число
```

```
finish
```

; здесь уже не читаем

В потоке вывода будет:

2	-01 00 00 01 01 01 01 / 01 00
	-01 00 00 01 01 01 00 . 01
10	-07 09 / 02
	-07 08 . 05