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 70-ричная система счисления. Для кодирования цифры в этой системе исчисления применяется алфавит:

```
[0..9][A..Z][a..z][!,@,#,$,%,^,&,*]
```

Здесь '*' соотвествует числу 69 в 10-ичной системе исчисления.

1 Калькулятор чисел с плавающей точкой

В аргументах командной строки программе передаются следующие параметры:

- **epsilon** — такой что: $0 \leq epsilon \leq 1$, 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 опрерации подразумевается следующий формат операции:

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

Здесь **result** — это скрытая переменная которую хранит калькулятор для хранения результата всех выполненых 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 70 включительно записанное в основании 10.

В тексте могут встречаться коментарии, пустые строки: строки состоящие из произвольного числа пробельных символов в том числе ни одного, строки по сути состоящие только из коментариев. Коментарий — 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 коментарий.

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 как-то ответ, иначе сообщение об ошибке и 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 4-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

```
+ 70:1#.E ; число 1
* 2:-0.1   ; число 2

finish
; здесь уже не читаем
```

В потоке вывода будет:

```
2  -1000011.00011
10 -67.1
```

2 Калькулятор рациональных чисел

В аргументах командной строки программе передаются следующие параметры:

- **epsilon** — такой что: $0 \leq epsilon \leq 1$, 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 рациональным числом **num** и двуместной операцией производимой с ним. Операции следующие: '+', '-', '*', '/'. Для опрерации подразумевается следующий формат операции:

```
result := result − num;
```

Здесь **result** — это скрытая переменная которую хранит калькулятор для хранения результата всех выполненых операций. Собственно значение результата выводится в конце работы программы.

Число указывается вместе со своей системой счисления. Указывается в формате:

```
система_счисления:целая_часть_числитель/натуральная_часть_знаменатель
```

Система счисления указывается как число от 2-х до 70 включительно записанное в основании 10. Число должно храниться в паре переменных: longint для числителя и longword для знаменателя. Необходимо проверять, попадание в диапозон при считывании.

В тексте могут встречаться коментарии, пустые строки: строки состоящие из произвольного числа пробельных символов в том числе ни одного, строки по сути состоящие только из коментариев. Коментарий — 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_числитель/натуральная_часть_знаменатель
```

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 коментарий.

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 как-то ответ, иначе сообщение об ошибке и 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 система счисления, затем с 4-ой позиции в строке само число результат в соответствующей системе счисления 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

```
+ 70:1#E/10; число 1
* 2:-1/10   ; число 2

finish
; здесь уже не читаем
```

В потоке вывода будет:

```
2  -1000011.00011
10 -67.1
```

3 Калькулятор целых чисел переменной битности

В аргументах командной строки программе передаются следующие параметры:

- **bit_wide** — число бит $0 < \text{bit\_wide} \leq 64$, которое отводится под хранение результата; 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 рациональным числом **num** и двуместной операцией производимой с ним. Операции следующие: '+', '-', '*', 'div', 'mod'. Для опрерации подразумевается следующий формат операции:

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

Здесь **result** — это скрытая переменная которую хранит калькулятор для хранения результата всех выполненых операций. Собственно значение результата выводится в конце работы программы.

Число указывается вместе со своей системой счисления и битностью. Указывается в формате:

```
система_счисления:ширина_в_битах:значение_числа
```

Система счисления указывается как число от 2-х до 70 включительно записанное в основании 10. Число должно храниться в типе int64. Необходимо проверять, попадание в дипозон ограниченный указанной битностью при считывании и на переполнение при выполнении операций.

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

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

```
операция система_счисления:ширина_в_битах:значение_числа
```

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 коментарий.

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 как-то ответ, иначе сообщение об ошибке и завершиться.

Скрытая переменная калькулятора **result** должна иметь тип int64 и быть инициализирована нулём. При вычислениях нигде не допустимо переполнение битности выданной результату. В случае переполнения печатать последний не переполненый промежуточный результат вычисления, в формате указанном далее, выдадавать сообщение о перепонении и завершать программу.

Результат должен выводиться для каждой из указанных в аргументах системы счисления следующим образом: на каждой строчке сперва система счисления, затем с 4-ой позиции в строке само число результат в соответствующей системе счисления.

3.1 Пример

А вот здесь его не будет, поскольку практически аналогично предыдущим вариантам задания.