Язык древних русов

Для счёта древних русов используется служебный сивол \$, а сами цифры выглядат как

0. $\$NOL\$$
1. $\$CELKOVIY\$$
2. $\$POLUSHKA\$$
3. $\$CHETVERTUSHKA\$$
4. $\$OSMUSHKA\$$
5. $\$PUDOVICHOK\$$
6. $\$MEDYACHOK\$$
7. $\$SEREBRYACHOK\$$
8. $\$ZOLOTNICHOK\$$
9. $\$DEVYATICHOK\$$

Числа из них порожадются стандартно, обозначим число за $number$.

Для логических значений выделен служебный сивол ? – $?PRAVDA?$ и $?KRIVDA?$. Обозначим лог. занчение за $bool$.

Идентификаторы – слова из $['r','u','s','R','U','S']$, причём сначала идут маленькие, затем заглавные символы:

$$S_0 \to @S@$$
$$S \to' r'|'u'|'s'|'R'|'U'|'S'$$
$$S \to (|'r'|'u'|'s')S$$
$$S \to S('R'|'U'|'S')$$

Примеры : "$@uuuURR@$", $@SRU@$, $@rusRUS@$. Обозначим слово из языка этой грамматики за $ident$. Логический идентификатор log_ident – то же самое, только обособляется символом !.
Каждый идентификатор соотвуетсвует одной глобальной переменной соответствующего типа. Изначально каждая переменная равна 0 или $?KRIVDA?$.
Введём понятие выражений – num_expr и log_expr.

Операторы – стандартные , со стандартными ассоциативностями/приоритетами.
Но! Унарный минус выглядит как $\_ - \_$ и их может быть сколько угодно, неассоциативен, имеет высший приоритет. Унарный плюс не предусмотрен. Разделим операторы на три типа $arithm\_op$ – арифметические, $comp\_op$ – операторы сравнение, log_op – логические.

Числовые выражения num_expr строятся следующим образом:

$S_0 \rightarrow : S :$
$S \rightarrow number|ident|S\, arithm\_op\, S$

Пример: : $\$PUDOVICHOK\$ * (@ruu@ + @S@ * \$NOL\$) :$

Логические выражения num_expr строятся следующим образом :

$S_0 \rightarrow ; S;$
$S \rightarrow bool|log\_ident|S\, log\_op\, S|num\_expr\, comp\_op\, num\_expr$

Пример: ; : $\$PUDOVICHOK\$ * (@ruu@ + @S@ * \$NOL\$) : ||(!ruu!\&\&?KRIVDA?);$

Блоки кода должны начинать на служебный сивол { и заканчиваться на }. Внутренность же должна начинаться на ключевое слово и быть в должной конструкции, задаваемой оным. Обозначим за *blok*.

Ключевые слова – обособляются служебным сиволов #:

- $\#PUSTO\#$ – пустой блок, кроме ключего слова ничего не должно быть.
- $\#ROBIT\#$. Синтаксис:
  $\#ROBIT\#blok1\ blok2\ blok3..$ – выполняет последовательно блоки.
- $\#ZVYAZATI\#$ и $\#LOGZVYAZATI\#$. Синтаксис:
  $\#ZVYAZATI\#\ ident\ num\_expr$ и $\#LOGZVYAZATI\ log\_ident\ log\_expr$ – кладёт в переменную значение данного выражения.
- $\#KOLI\#$, $\#TADI\#$, $\#PO-INOMU\#$. Синтаксис :
  $\#KOLI\#\ log\_expr\ \#TADI\#\ blok\ \#PO-INOMU\#\ blok$ – выполняет один из двух блоков, в зависимости от значения log_expr.
- $\#PAKUL\#$. Синтаксис:
  $\#PAKUL\#\ log\_expr\ blok$ – выполняет блока, пока верно условие log_expr.
- $\#NAPISATNABERESTU\#$ и $\#LOGNAPISATNABERESTU\#$. Синтаксис:
  $\#NAPISATNABERESTU\#ident$ и $\#LOGNAPISATNABERESTU\#log\_ident$ – выводит на бересту значение переменной.

Для корректности программы достаточно одно условие – она начинается и заканичвается в одном блоке $\#ROBIT\#$

Пример: {$\#ROBIT\#$
{
$\#ZVYAZATI\#@USSR@ : \$CELKOVIY\$\$DEVYATICHOK\$\$DEVYATICHOK\$\$CELKOVIY\$+$
$\_ - \_\$CELKOVIY\$\$DEVYATICHOK\$\$POLUSHKA\$\$POLUSHKA\$:$
}
{
$\#KOLI\#; @USSR@ > \$CELKOVIY\$\$NOL\$\$NOL\$;$
$\#TADI\#\{\#LOGZVYAZATI\#!USSR!; ?PRAVDA?;\}$
$\#PO-INOMU\#\{\#PUSTO\#\}\}$
}
{

$\#PAKUL\#; !USSR!; \{$
$\#NAPISATNABERESTE\# : \$NOL\$:$
$\}\}$
$\}$

Ах да, все пробелы, табы и перенос строк в документации исключительно потому что тех не даёт делать иначе. В языке их нет.