[6.9] Convert floating-point numbers to exact fractions

The usual method to convert a floating-point number *n* to an exact fraction is to use *exact(n)*. However, this function will only work for numbers smaller that the 14-digit precision of the 89/92+. For example, exact(1.234567890123456) results in 6172839450617/5000000000000, which is actually 1.23456789012; the ...3456 from the original number has been lost.

This program can be used to convert arbitrarily long floating point number to exact fractions.

```
stoexact(str)
Func
©Convert string argument to exact number

Local b,t,s

sign(expr(str))→s

if inString(str,"-")=1
right(str,dim(str)-1)→str

inString(str,".")→b

If b≠0 then
 right(str,dim(str)-b)→t
 Return s*(exact(iPart(expr(str)))+exact(expr(t))/10^(dim(t)))

else
 return exact(expr(str))
EndIf

EndFunc
```

The argument is passed as a string, for example

```
stoexact("1.234567890987654321")
```

returns 1234567890987654321/10000000000000000000.

This function works for positive and negative arguments, but does not support exponential notation.

(credit for original idea & core code to Kenneth C. Arnold)