NAME

icalc - interactive calculator

SYNOPSIS

icalc [**file**]

DESCRIPTION

Icalc is a algebraic calculator designed primarily for interactive use. Each formula definition *file* is read and compiled from the RADIANCE library where it is found. The standard input is then read, expressions are evaluated and results are sent to the standard output. If a newline is escaped using a backslash, input is continued on the next line.

An expression contains real numbers, variable names, function calls, and the following operators:

+ - * / ^

Operators are evaluated left to right, except '^', which is right associative. Exponentiation has the highest precedence; multiplication and division are evaluated before addition and subtraction. Expressions can be grouped with parentheses. Each result is assigned a number, which can be used in future expressions. For example, the expression ($3*10) is the result of the third calculation multiplied by ten. A dollar sign by itself may be used for the previous result. All values are double precision real.

In addition, variables and functions can be defined by the user. A variable definition has the form:

var = expression ;

Any instance of the variable in an expression will be replaced with its definition. A function definition has the form:

func(a1, a2, ..) = expression ;

The expression can contain instances of the function arguments as well as other variables and functions. Function names can be passed as arguments. Recursive functions can be defined using calls to the defined function or other functions calling the defined function.

To define a constant expression, simply replace the equals sign ('=') with a colon (':') in a definition. Constant expressions are evaluated only once, the first time they are used. This avoids repeated evaluation of expressions whose values never change. Ideally, a constant expression contains only numbers and references to previously defined constant expressions and functions. Constant function definitions are are replaced by their value in any expression that uses them with constant arguments. All predefined functions and variables have the constant attribute. Thus, "sin(PI/4)" in an expression would be immediately replaced by ".707108" unless sin() or PI were redefined by the user. (Note that redefining constant expressions is not a recommended practice!)

A variable or function's definition can be displayed with the '?' command:

? name

If no name is given, all definitions are printed. The '>' command writes definitions to a file:

> file

Similarly, the '<' command loads definitions.

The following library of predefined functions and variables is provided:

PI   the ratio of a circle's circumference to its diameter.

if(cond, then, else)
if cond is greater than zero, then is evaluated, otherwise else is evaluated. This function is necessary for recursive definitions.

select(N, a1, a2, ..)
> return aN (N is rounded to the nearest integer). This function provides array capabilities. If *N* is zero, the number of available arguments is returned.

rand(x)   compute a random number between 0 and 1 based on x.

min(a1, a2, ..)
> return the minimum value from a list of arguments.

max(a1, a2, ..)
> return the maximum value from a list of arguments.

floor(x)   return largest integer not greater than x.

ceil(x)   return smallest integer not less than x.

sqrt(x)   return square root of x.

exp(x)   compute e to the power of x (e approx = 2.718281828).

log(x)   compute the logarithm of x to the base e.

log10(x)   compute the logarithm of x to the base 10.

sin(x), cos(x), tan(x)
> trigonometric functions.

asin(x), acos(x), atan(x)
> inverse trigonometric functions.

atan2(y, x) inverse tangent of y/x (range –pi to pi).

ENVIRONMENT
> RAYPATH   the directories to check for auxiliary files.

AUTHOR
> Greg Ward

SEE ALSO
> ev(1), rcalc(1), tabfunc(1)