NAME

expr − evaluate arguments as an algebraic expression

SYNOPSIS

expr arg ...

DESCRIPTION

The arguments are taken as an expression. After evaluation, the result is written on the standard output. Terms of the expression must be separated by blanks. Characters special to the Shell, i.e., ‘*’, ‘|’, ‘&’, ‘(’, and ‘)’, must be escaped.

The operators and keywords are listed below. Characters that need to be escaped are preceded by ‘\’. The list is in order of increasing precedence, with equal precedence operators grouped within ‘{ }’ symbols.

expr \| *expr*

expr \& *expr*

expr { +, − } *expr*

expr { *, /, % } *expr*

substr *expr expr expr*

length *expr*

index *expr expr*

\(*expr* \)

The result of *substr* is that portion of the first expression (possibly null) which is defined by the offset (second expression, starting at ‘1’) and the span (third expression). A large span value can be given to obtain the remainder of the string.

Length returns the length in characters of the expression that follows.

Index searches the first expression for the first character that matches a character from the second expression. It returns the character position number if it succeeds, or ‘0’ if it fails.

The *expr* command is handy with Shell variables. For example:

```
expr substr xxx$a "(" length xxx$a − 2 ")" 3 | = b
```

assigns the last three characters of the Shell variable *$a* into the variable *$b*.

Note that ‘0’ is returned to indicate a zero value, rather than the null string. Strings containing blanks or other special characters should be quoted.

DIAGNOSTICS

Grumbles from *yacc*(I) for syntax violations.
"Non-numeric argument" if the argument needs to be, but is not, an integer.