The basic vocabulary consists of basic symbols classified into letters, digits, and special symbols. The special symbols are operators and delimiters:

```
+ ; ( and end nil set
- ) array file not then
* = [ begin for of to
/ <> ] case function or type
:= < { const goto packed until
. <= ] div if procedure var
. >= } do in program while
; > . downto label record with
else mod repeat
```

Word-delimiters (or reserved words) are normally underlined in the hand-written program to emphasize their interpretation as single symbols with fixed meaning. The programmer may not use these words in a context other than that explicit in the definition of Pascal; in particular, these words may not be used as identifiers. They are written as a sequence of letters (without surrounding escape characters).

The construct:

```
{<any sequence of symbols not containing "}">
```

may be inserted between any two identifiers, numbers, or special symbols. It is called a comment and may be removed from the program text without altering its meaning. The symbols { and } do not occur otherwise in the language, and when appearing in syntactic descriptions, they denote meta-symbols like | and :: . (On systems where the curly brackets are unavailable, the character pairs (* and *) are used in their place.)

Identifiers are names denoting constants, types, variables, procedures, and functions. They must begin with a letter, which may be followed by any combination and number of letters and digits. Although an identifier may be very long, implementations may impose a limit as to how many of these characters are significant. Implementations of Standard Pascal will always recognise the first 6 characters of an identifier as significant. That is, identifiers denoting distinct objects should differ in their first 6 characters.
examples of legal identifiers:
sum root3 pi h4g x
thisisaverylongbutneverthelesslegalidentifier
thisisaverylongbutprobablythesameidentifierasabove

illegal identifiers:
3rd array level 4 root -3

Certain identifiers, called standard identifiers, are predefined (e.g. sin, cos). In contrast to the word-delimiters (e.g. array), one is not restricted to this definition and may elect to redefine any standard identifier, as they are assumed to be declared in a hypothetical block surrounding the entire program block.

Decimal notation is used for numbers. The letter E preceding the scale factor is pronounced as "times 10 to the power of". The syntax of unsigned numbers is summarized in figure 1.b.

Note that if the number contains a decimal point, at least one digit must precede and succeed the point. Also, no comma may occur in a number.

unsigned numbers:
3 03 6272844 0.6 5E-8 49.22E+08 1E 10