Binary digits	As decimal number	As processor instruction	As letter of the alphabet
0100 0011	67	Move data from one register to another	C


LGP-30
COND STOP
START READ
STOP READ
PUNCH ON
TAPE FEED
CODE DELETE
MAN
JUMP
TAB
BACK SPACE
COLOR SHIFT
LOWER CASE
UPPER CASE
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
ON
POWER

$$f(x)=\frac{1}{2\pi}\int_{\infty}^{-\infty} e^{ipx}\left(\int_{\infty}^{-\infty} e^{-ipx} f(\alpha)\,d\alpha\right)dp$$

$$f(x)=\frac{1}{2\pi}\int_{\infty}^{-\infty} e^{ipx}\left(\int_{\infty}^{-\infty} e^{-ipx} f(\alpha)\,d\alpha\right)dp$$

```
      DIMENSION A(11)
      FUN(T) = SQRT(ABS(T)) + 5.)*T**3
      READ (5,1) A
 1    FORMAT(5F10.2)
      DO 10 J = 1, 11
         I = 11 - J
         Y = FUN(A(I+1))
         IF (400.0-Y) 4, 8, 8
 4            WRITE (6,5) I
 5            FORMAT(I10, 10H TOO LARGE)
         GO TO 10
 8            WRITE(6,9) I, Y
              FORMAT(I10, F12.6)
10    CONTINUE
      STOP
      END
```

Fortran (Formula Translation)

$$V_N = \{S, S_1, S_2, A, \bar{A}, B, \bar{B}, C, D, E, F\},$$

and the following rules:

(I) (a) $S \rightarrow CDS_1S_2F$

(b) $S_2 \rightarrow S_2S_2$

(c) $\begin{cases} S_2F \rightarrow BF \\ S_2B \rightarrow BB \end{cases}$

(d) $S_1 \rightarrow S_1S_1$

(e) $\begin{cases} S_1B \rightarrow AB \\ S_1A \rightarrow AA \end{cases}$

<identifier> ::=
    <letter>
  | <identifier> <letter>
  | <identifier> <digit>

<block> ::= { <statement list> }

<statement list> ::=
    <statement>
  | <statement list> <statement>

The following sentence is false.

The following sentence is false.

Rabbits are a type of fish.

The following sentence is false.

The Earth is flat.

The following sentence is false.

The preceding sentence is true.


The following sentence is false.
The preceding sentence is true.

Ancestry: Person A is an ancestor of Person B

If Person A is a parent of Person B

If Person A is a parent of a parent of Person B

If Person A is a parent of a parent of a parent of Person B

If Person A is a parent of a parent of a parent of a parent of . . .

The function "Ancestor(A,B)" returns the value 'True' if A is an ancestor of B and returns 'False' if not.

```
DEFINE Ancestor(A,B):

IF B = "Adam"
THEN RETURN 'False'

OTHERWISE

IF A = Parent(B) OR Ancestor(A,Parent(B)) = 'True'
THEN RETURN 'True'
```

The function "Ancestor(A,B)" returns the value 'True' if A is an ancestor of B and returns 'False' if not.

```
DEFINE Ancestor(A,B):

IF B = "Adam"
THEN RETURN 'False'

OTHERWISE

IF A = Parent(B) OR Ancestor(A,Parent(B)) = 'True'
THEN RETURN 'True'
```

The function "Ancestor(A,B)" returns the value 'True' if A is an ancestor of B and returns 'False' if not.

```
DEFINE Ancestor(A,B):

IF B = "Adam"
THEN RETURN 'False'

OTHERWISE

IF A = Parent(B) OR Ancestor(A,Parent(B)) = 'True'
THEN RETURN 'True'
```

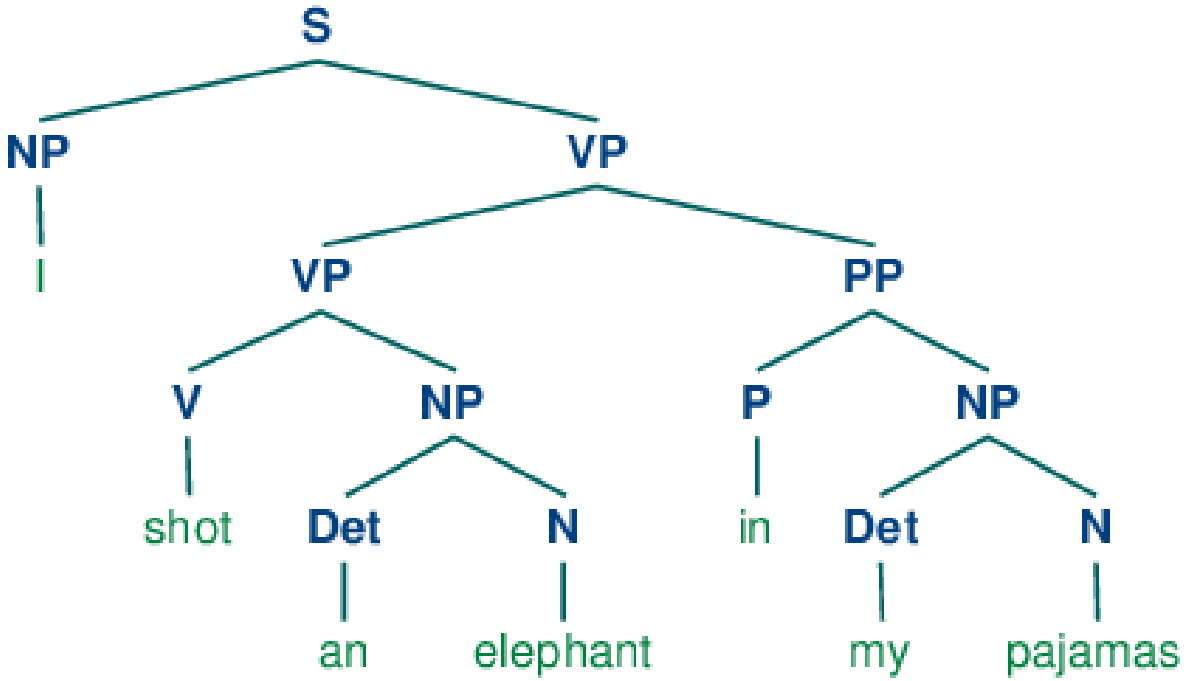
The function "Ancestor(A,B)" returns the value 'True' if A is an ancestor of B and returns 'False' if not.

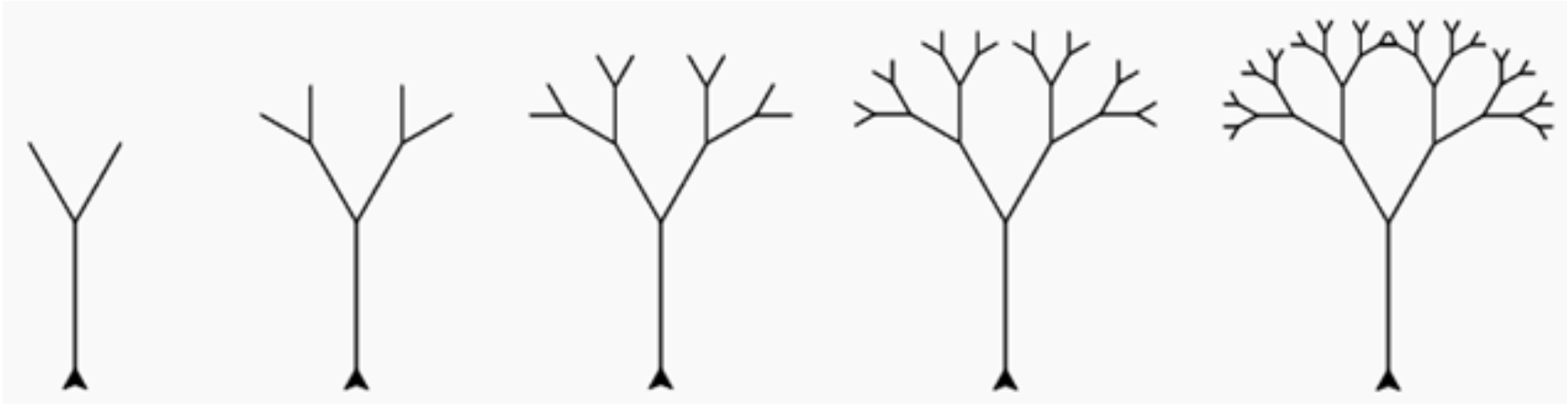
```
DEFINE Ancestor(A,B):

IF B = "Adam"
THEN RETURN 'False'

OTHERWISE

IF A = Parent(B) OR Ancestor(A,Parent(B)) = 'True'
THEN RETURN 'True'
```


S
NP
VP
I
VP
PP
V
NP
P
NP
shot
Det
N
in
Det
N
an
elephant
my
pajamas



Functions

#1 Double-it(x) = x times 2

#2 Square-it(x) = x times x

...

#500 Deduct-a-fifth(x) = x times ⅘

#501 Cube-it(x) = x times x times x

...

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

…

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

…

One-more-than(*x*) = Function #*x* plus 1

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

…

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

…

One-more-than(2) = Function #2 plus 1

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

…

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

…

One-more-than(2) = 2 times 2 plus 1

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

…

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

…

One-more-than(500) = Function #500 plus 1

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

…

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

…

One-more-than(500) = 500 times ⅘ plus 1

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

...

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

...

#605 One-more-than = Function #*x* plus 1

...

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

…

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

…

#605 One-more-than = Function #*x* plus 1

…

Q. What is the value of One-more-than(605)?

Functions

#1 Double-it(*x*) = *x* times 2

#2 Square-it(*x*) = *x* times *x*

...

#500 Deduct-a-fifth(*x*) = *x* times ⅘

#501 Cube-it(*x*) = *x* times x times *x*

...

#605 One-more-than = Function #*x* plus 1

...

Q. What is the value of One-more-than(605)?

A. One-more-than(605) = One-more-than(605) + 1

Enter Hamlet a footeman in haste.

Ham. What Coachman? my Ladyes Coach for shame; her ladiships ready to come downe;

Enter Potkinn, a Tankerd bearer.

Pot. Sfoote *Hamlet;* are you madde? whether run you now you should brushe vp my olde Mistresse?

(George Chapman, Ben Jonson, and John Marston *Eastward Hoe*, STC 4970, 1605)

It is a truth not generally acknowledged that, in most discussions of works of English fiction, we proceed as if a third, two-fifths, a half of our material were not really *there*.

(John Burrows *Computation into Criticism: A Study of Jane Austen's Novels and an Experiment in Method* (Oxford: Clarendon Press, 1987) p. 1)

Nothing amuses more harmlessly than computation, and nothing is oftener applicable to real business or speculative inquiries. A thousand stories which the ignorant tell, and believe, die away at once, when the computist takes them in his gripe.

(Samuel Johnson *Johnsoniana* Ed. by J. Wilson Croker (London: John Murray, 1836) "174. Computation")