ALGORITHM 34
GAMMA FUNCTION
M. F. Lipp
RCA Digital Computation and Simulation Group,
Moorstown, New Jersey

real procedure Gamma (x) ; real x ;
comment This procedure generalizes the recursive factorial routine, finding \( \Gamma(1+x) \) for reasonable values of x. Accuracy vanishes for large \( x(x) > 10 \) and for negative x with small fractional parts. For x being a negative integer the impossible value zero is given;
begin test: if x < 0 then go to minus else if x < 1 then
begin integer i ; real y ; array a [1:8] ;
a[1] := -.57719955 ;
a[2] := .59820566 ;
a[3] := -.91820606 ;
a[5] := -.75570408 ;
a[6] := -.98820536 ;
a[7] := -.8705694 ;
a[8] := 1.0 ;
y := a [1] ;
for i := 1 step 1 until 8 do y := y \times x + a [i] ;
Gamma := y end hastings
else Gamma := \times Gamma (x-1) ; go to endgam;
minus: if x = -1 then Gamma := 0 else
Gamma := Gamma (x+1) / x ;
endgam : end gam

REMARK ON ALGORITHM 34
GAMMA FUNCTION [M. F. Lipp, Comm. ACM 4 (Feb. 1961)]
Margaret L. Johnson and Ward Sangren

The coefficients used in the calculation of the Hastings polynomial are used in reverse order. The algorithm should have
\begin{align*}
a[1] & = -.57719955 ; \\
a[2] & = .59820566 ; \\
a[3] & = -.91820606 ; \\
a[4] & = .48219932 ; \\
a[5] & = -.75570408 ; \\
a[6] & = -.98820536 ; \\
a[7] & = -.8705694 ; \\
a[8] & = 1.0 ; \\
y & = .0358634 ;
\end{align*}
for i := 1 step 1 until 8 do y := y \times x + a [i] ;
Further, since Gamma (x) = \Gamma(1+x), the divisor x in the statement labeled minus should be x+1.

REMARKS ON:
ALGORITHM 34 [S14]
GAMMA FUNCTION
[M. F. Lipp, Comm. ACM 4 (Feb. 1961), 106]
ALGORITHM 54 [S14]
GAMMA FUNCTION FOR RANGE 1 TO 2
[John R. Herndon, Comm. ACM 4 (Apr. 1961), 180]
ALGORITHM 80 [S14]
RECIPECAL GAMMA FUNCTION OF REAL

ARGUMENT
[William Holsten, Comm. ACM 5 (Mar. 1962), 166]
ALGORITHM 221 [S14]
GAMMA FUNCTION
[Walter Gautschi, Comm. ACM 7 (Mar. 1964), 143]
ALGORITHM 291 [S14]
LOGARITHM OF GAMMA FUNCTION
[M. C. Pike and I. D. Hill, Comm. ACM 9 (Sept. 1966), 684]

M. C. Pike and I. D. Hill (Reed. 12 Jan. 1966)
Medical Research Council’s Statistical Research Unit,
University College Hospital Medical School,
London, England

Algorithms 34 and 54 both use the same Hastings approximation, accurate to about 7 decimal places. Of these two, Algorithm 54 is to be preferred on grounds of speed.

Algorithm 80 has the following errors:
(1) \( R_{GAM} \) should be in the parameter list of \( RGR \).
(2) The lines
\begin{align*}
\text{if } x = 0 \text{ then begin } RGR := 0 \text{ ; go to EXIT end and}
\text{if } x = 1 \text{ then begin } RGR := 1 \text{ ; go to EXIT end}
\end{align*}
should each be followed either by a semicolon or preferably by an else.
(3) The lines
\begin{align*}
\text{if } x = 1 \text{ then begin } RGR := 1/ y \text{ ; go to EXIT end and}
\text{if } x < -1 \text{ then begin } y := y \times x \text{ ; go to CC end}
\end{align*}
should each be followed by a semicolon.
(4) The lines
\begin{align*}
\text{if } x = -1 \text{ then begin } RGR := 0 \text{ ; go to EXIT end and}
\text{if } x > -1 \text{ then begin } RGR := R_{GAM}(x) \text{ ; go to EXIT end}
\end{align*}
should be separated either by else or by a semicolon and this second line needs terminating with a semicolon.
(5) The declarations of integer i and real array B[0:13] in \( R_{GAM} \) are in the wrong place; they should come immediately after begin real z;

With these modifications (and the replacement of the array B in \( R_{GAM} \) by the obvious nested multiplication) Algorithm 80 ran successfully on the ICT Atlas computer with the ICT Atlas ALGOL compiler and gave answers correct to 10 significant digits.

On grounds of speed that a choice should be made between them.
Algorithms 80 and 221 take virtually the same amount of computing time, being twice as fast as 291 at \( x = 1 \), but this advantage decreases steadily with increasing \( x \) so that at \( x = 7 \) the speeds are about equal and then from this point on 291 is faster—taking only about a third of the time at \( x = 25 \) and about a tenth of the time at \( x = 78 \). These timings include taking the exponential of log-gamma.

For many applications a ratio of gamma functions is required (e.g. binomial coefficients, incomplete beta function ratio) and the use of algorithm 291 allows such a ratio to be calculated for much larger arguments without overflow difficulties.