ALGORITHM 34
GAMMA FUNCTION

M. F. Lipp
RCA Digital Computation and Simulation Group, Moorestown, New Jersey

```
real procedure Gamma (x) ; real x ;
comment This procedure generalizes the recursive factorial
  routine, finding Γ(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] := -.57719165 ;
            a [2] := .98820589 ; a [3] := -.89705694 ;
            a [4] := .91820686 ;
            a [5] := -.75670408 ; a [6] := .48219939 ;
            a [7] := -.19352782 ;
            a [8] := .03586834 ; y := a [1];
          for i := 2 step 1 until 8 do y := y × x + a [i] ;
          Gamma := y end hastings
          else Gamma := x × 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
Computer Applications, Inc., San Diego, Calif.

The coefficients used in the calculation of the Hasting's polynomial are used in reverse order. The algorithm should have

```
a[1]=-.19352782;  a[2]=.48219939;  a[3]=-.75670408;
a[4]=.91820686;   a[5]=-.89705694; a[6]=.98820589;
a[7]=-.57719165;  a[8]=1.0;
y=.03586834;
for i := 1 step 1 until 8 do y := y×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]
RECIPROCAL 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 (Recd. 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) *RGAM* should be in the parameter list of *RGR*.
(2) The lines

if $x = 0$ **then begin** $RGR := 0$; **go to** *EXIT* **end**

and

if $x = 1$ **then begin** $RGR := 1$; **go to** *EXIT* **end**

should each be followed either by a semicolon or preferably by an **else**.
(3) The lines

if $x = 1$ **then begin** $RGR := 1/y$; **go to** *EXIT* **end**

and

if $x < -1$ **then begin** $y := y \times x$; **go to** *CC* **end**

should each be followed by a semicolon.
(4) The lines

BB: **if** $x = -1$ **then begin** $RGR := 0$; **go to** *EXIT* **end**

and

if $x > -1$ **then begin** $RGR := RGAM(x)$; **go to** *EXIT* **end**

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 *RGAM* are in the wrong place; they should come immediately after

begin real z;

With these modifications (and the replacement of the array B in *RGAM* 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.

Algorithms 80, 221 and 291 all work to an accuracy of about 10 decimal places and to evaluate the gamma function it is therefore 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 *loggamma*.

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.