ALGORITHM 54
GAMMA FUNCTION FOR RANGE 1 TO 2

John R. Herndon
Stanford Research Institute, Menlo Park, California

```
real procedure  Q(x);  value x;  real x,
comment  This procedure computes Γ(x) for 1 ≦ x ≦ 2. This is
   a reference procedure for the more general gamma function
   procedure. Γ(x) = Q(x−1);
begin          Q := (((((((0.035868343 × x − 0.19352782) × x
                  + 0.48219939) × x − 0.75670408) × x
                  + 0.91820686) × x − 0.89705694) × x
                  + 0.98820589) × x − 0.57719165) × x + 1.0
end            Q;
```

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.