ALGORITHM 20
REAL EXPONENTIAL INTEGRAL .

S. PEAVY
National Bureau of Standards, Washington 25, D.C.

```
real procedure  Expint (x) ; real x ;
comment         -Ei(-x) = ∫_x^∞ (e^-u/u) du is computed for
                x > 0 by approximation formulas. For
                0 < x < 1 the approximation is from E. E.
                Allen, Note 169, MTAC 56, pg 240 (1954).
                The second approximation formula is for
                1 ≤ x < ∞ and is from C. Hastings, Jr.,
                "Approximations For Digital Computers"
                (Princeton University Press, Princeton,
                New Jersey, 1955). The absolute error
                ε(x) is |ε(x)| < 2 × 10^-7 for 0 < x < 1
                and |ε(x)| < 2 × 10^-8 for 1 ≤ x < ∞ ;
begin           real y, w, z ;
                if x < 1 then
                z := (((( .00107857 × x — .00976004) × x
                  + .005519968) × x — .24991055) × x
                  + .99999193) × x — .57721566 — ln(x)
                else begin
                y := ((( x + 8.5733287401) × x
                  + 18.059016973) × x + 8.6347608925) × x
                  + .2677737343 ;
                w := ((( x + 9.5733223454) × x
                  + 25.6329561486) × x
                  + 21.0996530827) × x + 3.9584969228 ;
                z := exp (—x) /x × (y/w) end
                Expint := z end
```

REMARK ON ALGORITHM 20
REAL EXPONENTIAL INTEGRAL (S. Peavy, *Comm. ACM*, October 1960)

S. PEAVY
National Bureau of Standards, Washington, D. C.

A printing error has been called to our attention by J. A. Beutler of E. I. duPont de Nemours and Co. Lines 15 through 17 of Algorithm 20 should read

```
z := ((((.00107857 × x − .00976004) × x
  + .05519968) × x − .24991055) × x
  + .99999193) × x − .57721566 − ln (x)
```

* Work supported by the U. S. Atomic Energy Commission.

CERTIFICATION OF ALGORITHM 20
REAL EXPONENTIAL INTEGRAL (S. Peavy, *Comm. ACM*, Oct. 1960)

WILLIAM J. ALEXANDER* and HENRY C. THACHER, JR.*
Argonne National Laboratory, Argonne, Illinois

Expint (x) was programmed for the LGP-30 computer, using both a 7S floating-point compiler (ACT III) and an 8S floating-point interpretive code (24.2). Constants given to more than 7S (or to 8S for the 24.2 program) were rounded to 7S (or 8S).

After changing the constant .005519968 to .05519968, both programs gave acceptable accuracy over the range tested.

The 8S (24.2) program was compared with the 9D values given for $-E_i(-x)$ in Mathematical Tables Project, *Tables of Sine, Cosine, and Exponential Integrals, Volume* II (1940) for the set of values x = 0.1(0.1)1.0(1.0)10.0. The largest discrepancy found was -16×10^{-8} for x = 0.1. For x greater than 1, all values tested were good to 8S.

For computing real values of the exponential integral, this algorithm is much faster than EKZ (Algorithm 13). For x < 1, the ratio of speeds was of the order of 20.

* Work supported by the U.S. Atomic Energy Commission.