ALGORITHM 42
INVERT

T. C. Wood
RCA Digital Computation and Simulation Group, Moorestown, New Jersey

```
procedure Invert (A) order: (n) Singular: (s) Inverse: (A1);
      array A, A1; integer n,s, value n;
comment This procedure inverts the square matrix A of order
   n by applying a series of elementary row operation to the matrix
   to reduce it to the identity matrix. These operations when
   applied to the identity matrix yield the inverse A1. The case
   of a singular matrix is indicated by the value s := 1;
begin      comment augment matrix A with identity matrix;
           array a[1:n, 1:2 × n]; integer i,j;
           for i := 1 step 1 until n do
           for j := 1 step 1 until 2 × n do
           if j ≦ n then a[i,j] := A[i,j] else
           if j = n+1 then a[i, j] := 1.0 else a [i,j] := 0.0;
           comment begin inversion;
           for i := 1 step 1 until n do
begin      integer k, ℓ, ind; j := ℓ := i; ind := s := 0;
      L1:  if a[ℓ,j] = 0 then
           begin ind := 1; if ℓ < n then begin ℓ := ℓ + 1;
           go to L1 end
                else begin s := 1; go to L2 end
              end;
           if ind = 1 then for k := 1 step 1 until 2 × n do
           begin real temp;
                temp := a[ℓ,k];
                a[ℓ,k] := a [i,k];
                a[i,k] := temp end k loop;
           for k := j step 1 until 2 × n do
              a[i,k] := a[i,k] / a[i,j];
           for ℓ := 1 step 1 until n do
           if ℓ ≠ i then for k := 1 step 1 until 2 × n do
              a[ℓ,k] := a[ℓ,k] - a[i,k] × a[ℓ,j];
           end i loop;
           for i := 1 step 1 until n do
           for j := 1 step 1 until n do
              A1[i,j] := a[i,n+j];
      L2:  end of procedure
```

CERTIFICATION OF ALGORITHM 42
INVERT (T. C. Wood, *Comm. ACM*, Apr., 1961)

Anthony W. Knapp and Paul Shaman
Dartmouth College, Hanover, N. H.

INVERT was hand-coded for the LGP-30 using machine language and the 24.0 floating-point interpretive system, which carries 24 bits of significance for the fractional part of a number and five bits for the exponent. The following changes were found necessary:

```
(a)  if j = n+1 then a[i, j] := 1.0 else a[i, j] := 0.0;
     should be
     if j = n+i then a[i, j] := 1.0 else a[i, j] := 0.0;

(b)  for k := j step 1 until 2 × n do
       a[i, k] := a[i, k]/a[i, j];
     should be
     for k := 2 × n step -1 until i do
       a[i, k] := a[i, k]/a[i, i];

(c)  if l ≠ i then for k := 1 step 1 until 2 × n do
       a[l, k] := a[l, k] - a[i, k] × a[l, j];
     should be
     if l ≠ i then for k := 2 × n step -1 until i do
       a[l, k] := a[l, k] - a[i, k] × a[l, i];
```

Given these changes, j becomes superfluous in the second i loop, and the other references to j may be changed to references to i.

INVERT obtained the following results:

The computer inverted a 17-by-17 matrix whose elements were integers less than ten in absolute value. When the matrix and its inverse were multiplied together, the largest nondiagonal element in the product was −.00003. Most nondiagonal elements were less than .00001 in absolute value.

INVERT was tested using finite segments of the Hilbert matrix. The following results were obtained in the 4 × 4 case:

16.005	−120.052	240.125	−140.082
−120.052	1200.584	−2701.407	1680.917
240.126	−2701.411	6483.401	−4202.217
−140.082	1680.920	−4202.219	2801.446

The exact inverse is:

16	−120	240	−140
−120	1200	−2700	1680
240	−2700	6480	−4200
−140	1680	−4200	2800

INVERT was also coded for the LGP-30 in machine language and the 24.1 extended range interpretive system. This system, which uses 30 significant bits for the fraction, obtained the following as the inverse of the 4 × 4 Hilbert matrix:

16.000	−120.001	240.001	−140.001
−120.001	1200.006	−2700.015	1680.010
240.001	−2700.016	6480.037	−4200.024
−140.001	1680.010	−4200.024	2800.016

The program coded in the 24.0 interpretive system successfully inverted a matrix consisting of ones on the minor diagonal and zeros everywhere else.

REMARKS ON ALGORITHM 42
INVERT [T. C. Wood, *Comm. ACM*, Apr. 1961]

P. Naur
Regnecentralen, Copenhagen, Denmark

INVERT cannot be recommended since it does not search for pivot and therefore will give poor accuracy. This is confirmed by the figures quoted by Knapp and Shaman in their certification [*Comm. ACM 4* (Nov. 1961), 498]. The results obtained by them using 30 significant bits for the fraction may be compared directly with those obtained using INVERSION II (Algorithm 120) and gjr with the Gier Algol system (see certification below). Inverting the 4 × 4 segment of the Hilbert matrix, the largest error in any element is found to be:

	Subscripts	*Error*
INVERT (Knapp and Shaman)	3,3	0.037
INVERSION II (see certification of	3,3	0.0306
gjr Alg. 120)	4,3	0.00010

In view of this basic shortcoming of Algorithm 42, it is unnecessary to report on other features of it.

CORRECTION TO EARLIER REMARKS ON ALGORITHM 42 INVERT, ALG. 107 GAUSS'S METHOD, ALG. 120 INVERSION II, AND gjr [P. Naur, *Comm. ACM*, Jan. 1963, 38–40.]

P. Naur
Regnecentralen, Copenhagen, Denmark

George Forsythe, Stanford University, in a private communication has informed me of two major weaknesses in my remarks on the above algorithms:

1) The computed inverses of rounded Hilbert matrices are compared with the exact inverses of unrounded Hilbert matrices, instead of with very accurate inverses of the rounded Hilbert matrices.

2) In criticizing matrix inversion procedures for not searching for pivot, the errors in inverting positive definite matrices cannot be used since pivot searching seems to make little difference with such matrices.

It is therefore clear that although the figures quoted in the earlier certification are correct as they stand, they do not substantiate the claims I have made for them.

To obtain a more valid criterion, without going into the considerable trouble of obtaining the very accurate inverses of the rounded Hilbert matrices, I have multiplied the calculated inverses by the original rounded matrices and compared the results with the unit matrix. The largest deviation was found as follows:

	Maximum deviation from elements of the unit matrix		
Order	*INVERSION II*	*gjr*	*Ratio*
2	$-1.49_{10}-8$	$-1.49_{10}-8$	1.0
3	$-4.77_{10}-7$	$-8.34_{10}-7$	0.57
4	$-9.54_{10}-6$	$-3.43_{10}-5$	0.28
5	$-7.32_{10}-4$	$-4.58_{10}-4$	1.6
6	$-1.61_{10}-2$	$-1.42_{10}-2$	1.1
7	$-5.78_{10}-1$	$-5.47_{10}-1$	1.1
8	$-1.20_{10}-2$	$-1.38_{10}1$	8.7
9	$-4.91_{10}1$	$-2.22_{10}1$	2.2

This criterion supports Forsythe's criticism. In fact, on the basis of this criterion no preference of INVERSION II or *gjr* can be made.

The calculations were made in the GIER ALGOL system, which has floating numbers of 29 significant bits.