ALGORITHM 94
COMBINATION
JEROME KURTZBERG
Burroughs Corp., Burroughs Laboratories, Paoli, Pa.

```
procedure COMBINATION (J, N, K);  value N, K;  integer
   array J;  integer N, K;
comment  This procedure generates the next combination of N
   integers taken K at a time upon being given N, K and the pre-
   vious combination. The K integers in the vector J(1) ⋯ J(K)
   range in value from 0 to N − 1, and are always monotonically
   strictly increasing with respect to themselves in input and
   output format. If the vector J is set equal to zero, the first·
   combination produced is N−K, ⋯ , N−1. That initial combina-
   tion is also produced after 0, 1, ⋯ , N−1, the last value in that
   cycle;
begin  integer B, L;
       B := 1:
mainbody:  if J(B)≧B then begin A := J(B) − B−1;
              for L := 1 step 1 until B do J(L) := L + A;
              go to exit end;
              if B = K then go to initiate;
              B := B + 1; go to mainbody;
initiate:     for B := 1 step 1 until K do J(B) := N − K − 1 + B
exit:         end COMBINATION
```

CERTIFICATION OF ALGORITHM 94
COMBINATION [J. Kurtzberg, *Comm. ACM*, June 1962]
RONALD W. MAY
University of Alberta, Calgary, Alberta, Canada

Algorithm 94 was translated into FORTRAN for the IBM 1620 and run successfully with no corrections. The variable A, however, has not been declared.

CERTIFICATION OF ALGORITHM 94
COMBINATION [J. Kurtzburg, *Comm. ACM*, June, 1962]
R. E. GRENCH*
Reactor Eng. Div., Argonne National Laboratory, Argonne, Ill.
   * Work supported by U. S. Atomic Energy Commission
   Four changes were required in the algorithm.
1. The last sentence in the comment should read: That initial combination is also produced after 0, 1, ⋯ , $K−1$, the last value in that cycle;
2. The integer A was declared;
3. Parentheses were replaced by brackets in the subscript expressions;
4. A semicolon was inserted at the end of the initiate statement.
   After the above changes were made the body of Algorithm 94 was tested on an LGP-30 computer using the Dartmouth College ALGOL-30 translator. The body tested satisfactorily and the time required to generate one J when $K = 5$ and $N = 15$ was 30 seconds.
   Various tests should be included if this algorithm is to be used as a procedure. These tests might include a statement to check if $K > N$ and if the initial value of J is correct These two possibilities were investigated and it was found that improper J's are generated.