DECUS

PROGRAM LIBRARY

DECUS NO.   8-480a

TITLE   TWO SUBROUTINES FOR 8K FORTRAN
         1. INPUT
         2. RANDU and GAUSS

AUTHOR   Lars Palmer

COMPANY   A. B. Hassle
           Goteborg, Sweden

DATE   April 26, 1972

SOURCE LANGUAGE   SABR

INPUT

A relocable input routing for input in free format to 8K fortran programs

Usage:

CALL INPUT (IDEV, A, NR)

where

IDEV is device code for input (standard 8K fortran usage (device
     codes 1 and 2 only), for device code 0 see below)
A   is first position to store the incoming numbers in (Floating pt
     vector)
NR   is number of items to be input

Input is in a free format, the numbers are separated (and terminated) by ,
or return. Two terminators in sequence without intervening input will set
number=0.

Special operators are as follows:

Character	ASCII	Meaning
,	254	terminates number
cr	215	" " must be given before return to main program is effected (but see CTRL/Z)
.	256	decimal point
rub	377	removes last input before terminator is given
CTRL/U	225	removes input since last cr
↑	336	restarts input routine (removes all input)
CTRL/Z *	241	terminates input & returns to main program, all remaining numbers are set to 0
-	255	negative number
CTRL/C *	203	return monitor directly.

All other characters and illegal uses of the above (or excess figures after
     all numbers have been input) are ignored and not echoed on the
     teletype.

* ↑ Z, and ↑ C will function during high speed reader input.

Ex CALL INPUT (I, A(5), 10) will read from the teletype 10 numbers to be
stored in A(5) - A(15).

A call to input with device number 0 will set the referenced variables = 0.
This is a fast and echonomical (in core space) method of zeroing a vector
before use
e g CALL INPUT (0,A(5), 10)
will set A(5) - A(15) to zero using less core space than the equivalent do
loop.

The routine requires 4 pages but much core space is saved by omitting read statements and their FORMATS not to mention the simplicity of input as opposed to formated input.

The routine ignores the parity bit and can be used on all teletypes. If the routine is to be used in a non-PS8 environment so must some slight re-editing be done, change CALL 0,EXIT at location "TEXIT" to "HLT" or "RETURN INPUT" whichever suits you best.

RANDU and GAUSS, random number generators for 8K fortran.

Usage.

RANDU: CALL RANDU (A, X)
GAUSS: CALL GAUSS (AM, S, A, XN)

Parameters:

A   A should on first calling be contain any random floating point number. On return it contains a new f.p. number which should be used for following input. It should not be 0. Caution A should not be manipulated by the SABR rutines for f.p. numbers (except for original definition) as these routines normalize the number.

X   On return a random normal number between 0 and 1

AM   Mean of required population

S   Standard deviation of required population

XN   random normal number from the population (M, S)

Method:
RANDU is the routine in DECUS 5-25 rewritten in SABR. The final number is converted into a positive floating point number. The 36bit starting number for the random number generator is stored in A and transferred word by word (i.e. not as a f.p. number).

GAUSS uses the approximation to normal distributions (from RW Hamming Numerical Methods for scientists and engineers, McGrawhill 1962).
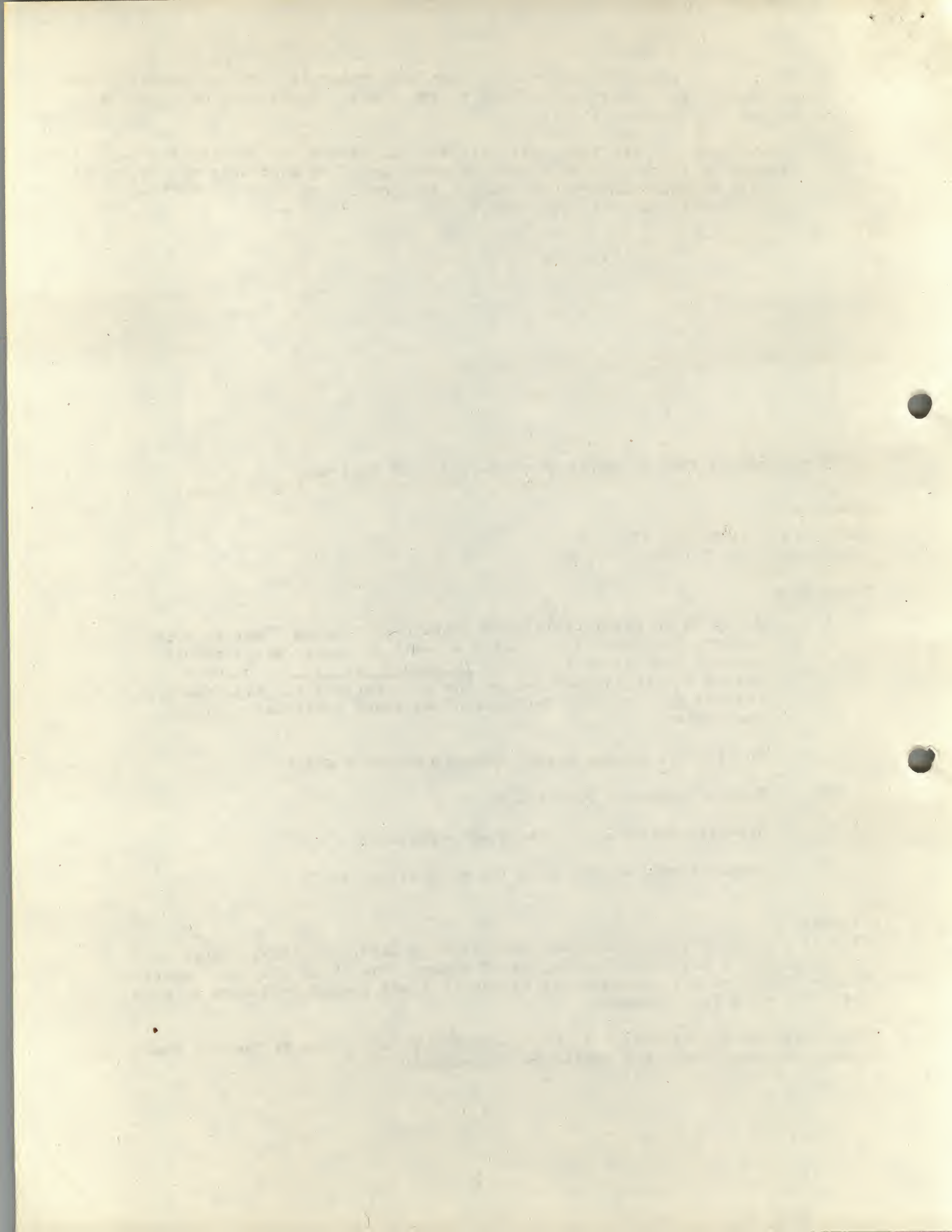
$$Y = \frac{\sum_{i=1}^{K} X_i - \frac{K}{2}}{\sqrt{K-12}}$$

with K = 12 this reduces to

$$Y = \sum_{i=1}^{12} X_i - 6$$

and adjusted to the required M and SD we have

$$XN = Y * S + AM$$

The X_i are generated by 12 calls to RANDU

Execution time (total, including transfer from calling fortran program)

RANDU approx: 30 msec
GAUSS "   150 msec

Storage 3 pages

3