APL\360

IBM RESEARCH

APL\360

APL\360 is a conversational terminal system that has been operating within IBM since the Fall of 1966. Conceived as an experimental system for exploring certain aspects of computer science, its purpose required that it operate under a realistic load in an environment that was not artificially constrained. To this end, the members of the IBM Research staff, and others, were encouraged to learn the APL language and use the APL system. It was not anticipated that it would have the impact that it did:

> In twenty months of regularly scheduled operation, clocking well over 100,000 terminal-hours of use, the availability of APL\360 has materially changed the computing habits of the Research organization.
>
> Heavy users of batch processing have turned to APL for on-line development of their algorithms.
>
> Many laboratory data reduction chores formerly done by batch operation or desk calculator are now executed in a timely way at local terminals, using locally written, stored APL programs.
>
> Automatic collection of experimental data has, in many instances, been put on-line to APL.
>
> Routine correspondence and technical papers are prepared at terminals, with the help of text-handling programs written in APL.
>
> Many professional staff members who formerly were indifferent to, or actively resisted, the use of computers, have become steady users of the APL system.

In addition to IBM Research personnel, use of the system was offered to other locations within the IBM Company, and it was also used experimentally in elementary and secondary schools and in universities. The general findings may be summarized as follows:

Although APL is easy for a beginner to learn, non-programmers (as well as programmers) often develop an interest in sophisticated use of the language, because of its analytical power and mathematical structure.

The primitive array operations of APL make it a good language for scientific problems, text handling, and general data processing, because arrays are fundamental to all of these applications.

The use of a powerful, readily accessible computational facility can materially change the quality and orientation of an academic course.

Other things being equal, acceptance of time-sharing as a general mode of operation is strongly dependent upon its reliability and availability -- regularly scheduled hours are essential, and the more the better, including nights and weekends.

System features. APL\360 is based upon APL, the language first defined by K. E. Iverson in A Programming Language (John Wiley, 1962). It is an interpretive time-sharing system that builds upon the array operations and structural integrity of APL to provide a running system with the following salient characteristics:

Simple, uniform rules of syntax

Use of common symbols for the ordinary arithmetic operations

Free-form decimal input

A large set of primitive operators

Use of defined functions (programs) with the same facility and syntactic variety as primitive operators

Automatic internal conversions of data representation, with full double-precision arithmetic (16 decimal digits) when required.

Very fast response

A library structure built around workspaces that hold both programs and data

An immediate-execution mode completely free of irrevelant keywords

A comprehensive, integrated set of system commands for managing workspaces and libraries and for other essential functions

Three levels of security; account numbers, workspaces, and programs can be individually locked against use or display

Visual fidelity between hard copy and transmitted entries, which ensures reproducibility of results

Succinct diagnostic reports

Hardware and performance. The APL\360 system, comprising a time-sharing supervisor and an APL interpreter, runs in concert with a modified DOS Version 3, which appears the same as the standard DOS release of the corresponding change level. During APL operation it requires an S/360 machine with at least an H-level memory, a transmission control unit, and three disk drives.

The particular configuration used in the original Research experiment is a Model 50 H with a 2703 Transmission Control Unit and a 2314 Direct Access Storage Facility. This system has 61 ports, and with about that number in use seems to be a well-balanced system with regard to relative usage of the central processor and input-output equipment.

APL\360 has run (or is now running) experimentally on S/360 Model 40 and Model 65 machines, and will run on all larger models. It will also run on an S/360 Model 44 equipped with the Trap and Emulate RPQ. It supports IBM 2741 and 2740-1 Communications Terminals, and 1050 Tele-processing Terminals with or without paper tape or punched-card equipment.

Average reaction time of the Model 50 system (i.e., time to respond to trivial requests from a terminal) peaks at about three-tenths of a second. With up to 40 ports in use, most such responses are essentially instantaneous; when fully loaded, there are occasional delays of as much as four seconds.

The time for serving non-trivial requests naturally varies according to the extent to which the CPU must be shared during the computation. Because the primitive operations of APL are defined on arrays, relatively little interpretive overhead is needed for many large computations, and the actual CPU time used for a typical computation may run from 20 to 50 times that for efficient compiled code; but the overall efficiency is likely to be comparable, if not superior, to batch processing in many applications if the usual compiling and loading times for batch work are taken into account. If debugging time is included, the advantage of interpretive APL becomes even greater.

Applications. The workspace-centered organization of APL\360 makes it very convenient to assemble application packages, because a collection of programs and data in a workspace moves as a coherent unit whenever the workspace containing it is stored or activated. Some of the packages that have been developed on the experimental systems so far are:

A comprehensive collection of functions for statistical calculations

A coordinate geometry package that includes a position plotter

A management information system for controlling the allocation of manufacturing resources

A text-editing and composing facility

An algebraic manipulator that simplifies polyomials and operates on matrices of algebraic expressions

A bookkeeping system for balancing accounts

Many collections of programs for numerical analysis of both real and complex functions

The IBM Research Model 50 APL\360 system serves more than 700 active users whose work ranges over many fields: from simple desk calculations to experimental investigations of semi-groups, from laboratory data reduction to theoretical physics, from the working out of small programming problems to the design and modelling of major systems, and from administering arithmetic drill to elementary-school children to use in graduate-level engineering courses.

The accompanying appendix gives examples of operator-computer dialogue and the corresponding functions which would take place during sample terminal exercises.

Appendix

SAMPLE TERMINAL EXERCISES

```
)1776
010)  19.32.36 07/03/68  JANET

   A P L \ 3 6 0

                                     FUNDAMENTALS

      3×4                            Entry automatically indented
12                                   Response not indented
      X←3×4                          X is assigned value of
                                          the expression
      X
12                                   Value of X typed out
      Y←¯5                           Negative sign for negative
                                          constants
      X+Y
7
      144E¯2                         Exponential form of constant
1.44
      P←1 2 3 4                      Four-element vector
      P×P                            Functions apply element by element
1  4  9  16
      P×Y                            Scalar applies to all elements
¯5  ¯10  ¯15  ¯20
      Q←'CATS'                       Character constant (4-element
      Q                                   vector)
CATS
      YZ←5                           Multi-character names
      Y_Z_1←5
      YZ+Y_Z_1
10
      3+4×5+6                        Correction by backspace
         v                               and linefeed
         +5+6
18
      X←3
      Y←4
      (X×Y)+4
16
      X×Y+4                          Executed from right to left
24
```

```
      X Y                            Entry of invalid expression
SYNTAX ERROR                         Shows type of error committed
      X Y                            Retypes invalid statement with
      ∧                                   caret where execution stopped
      XY                             Multi-character name (not X×Y)
VALUE ERROR
      XY                             XY had not been assigned a value
      ∧

                                     SCALAR FUNCTIONS

      4×3⌈5.1                        Dyadic maximum
20.4
      (4×3)⌈5.1
12
      4×⌈5.1                         Monadic ceiling
24
      X←⍳5                           Index generator function
      X
1  2  3  4  5
      ⍳0                             Empty vector
                                          prints as a blank line
      Y←5-X                          All scalar functions extend
      Y                                   to vectors
4  3  2  1  0
      X⌈Y
4  3  3  4  5
      X≤Y                            Relations produce
1  1  0  0  0                             logical (0 1) results
      ○1                             Pi×1
3.141592654
      ○÷1 2                          Pi÷1 2
3.141592654  1.570796327
      X←45 90
      ○X÷180                         Conversion of X to radians
0.7853981634  1.570796327
      1○1                            Sin 1
0.8418709848
      2○1 2                          Cos 1 2
0.5403023059  ¯0.4161468365
      3○1                            Tan 1
1.557407725
      ¯3○1                           Arctan 1
0.7853981634
      3○¯3○⍳7                        Tan Arctan 1 2 3 4 5 6 7
1  2  3  4  5  6  7
      Y←1 2
      4○Y                            (1+Y*2)*.5
1.414213562  2.236067977
      0○÷Y                           (1-÷Y*2)*.5
0  0.8660254038
      7○1 2                          Tanh 1 2
0.761594156  0.9640275801
      ¯7○7○1 2                       Arctanh Tanh 1 2
1  2
```

```
                                          DEFINED FUNCTIONS

        ∇Z←X F Y                          Header (2 args and result)
[1]     Z←((X*2)+Y*2)*.5                  Function body
[2]     ∇                                 Close of definition
        3 F 4                             Execution of dyadic function F
5
        P←7
        Q←(P+1)F P-1                      Use of F with expressions
        Q                                      as arguments
10
        4×3 F 4
20
        ∇B←G A                            G is the signum function
[1]     B←(A>0)-A<0                       A and B are local variables
[2]     ∇
        G 4
1
        G ¯6
¯1
        X←¯6
        G X
¯1
        ∇H A                              Like G but has no explicit result
[1]     P←(A>0)-A<0                       P is a global variable
[2]     ∇
        H¯6
        P
¯1
        Y←H¯6                             H has no explicit result
VALUE ERROR                                    and hence produces a value
        Y←H ¯6                                 error when used to right
          ∧                                    of assignment
        ∇Z←FAC N;I                        FAC is the factorial function
[1]     Z←1
[2]     I←0
[3]     L1:I←I+1                          L1 becomes 3 at close of def
[4]     →0×⍳I>N                           Branch to O (out) or to next
[5]     Z←Z×I
[6]     →L1                               Branch to L1 (that is, 3)
[7]     ∇
        FAC 3
6
        FAC 5
120
        T∆FAC←3 5                         Set trace on lines 3 and 5 of FAC
        X←FAC 3
FAC[3] 1                                  Trace of FAC
FAC[5] 1
FAC[3] 2
FAC[5] 2
FAC[3] 3
FAC[5] 6
FAC[3] 4
        T∆FAC←0                           Reset trace control
```

MECHANICS OF FUNCTION DEFINITION

```
        ∇G←M GCD N                 Greatest common divisor
[1]     G←N                             function based on the
[2]     M←M|N                           Euclidean algorithm
[3]     →4×M≠0
[4]     [1]G←M                     Correction of line 1
[2]     [4]N←G                     Resume with line 4
[5]     [1⎕]                       Display line 1
[1]     G←M
[1]     [⎕]                        Display entire GCD Function
      ∇ G←M GCD N
[1]     G←M
[2]     M←M|N
[3]     →4×M≠0
[4]     N←G
      ∇                            Close of display, not close of def
[5]     →1                         Enter line 5
[6]     ∇                          Close of definition
        36 GCD 44                  Use of GCD
4                                  4 is GCD of 36 and 44
        ∇GCD                       Reopen def (Use ∇ and name only)
[6]     [4.1]M,N                   Insert between 4 and 5
[4.2]   [⎕]                        Display entire function
      ∇ G←M GCD N
[1]     G←M
[2]     M←M|N
[3]     →4×M≠0
[4]     N←G
[4.1]   M,N                        Fraction stays until close of def
[5]     →1
      ∇                            End of display
[6]     ∇                          Close of definition
        36 GCD 44
8  36                              Iterations printed by
4  8                                    line 5 (was line 4.1)
4                                  Final result
        ∇GCD[⎕]∇                   Reopen, display, and close GCD
      ∇ G←M GCD N
[1]     G←M
[2]     M←M|N
[3]     →4×M≠0
[4]     N←G
[5]     M,N                        Line numbers have been
[6]     →1                              reassigned as integers
      ∇                            Close (Even number of ∇'s in all)
        ∇GCD                       Reopen definition of GCD
[7]     [5]                        Delete line 5 by linefeed
           ∧
        ∇                          Close definition
```

```
      ∇Z←ABC X                       A function to show line editing
[1]   Z←(33×Q+(R×5)-6                A line to be corrected
[2]   [1⎕9]                          Initiate edit of line 1
[1]   Z←(33×Q+(R×5)-6                Types line, stops ball under 9
         /    1 /1                   Slash deletes, digit inserts spaces
[1]   Z←(3×Q)+(T×5)-6                Ball stops at first new
[2]   ∇                                   space. Then enter )  T
      FAC 5                          FAC still defined
120
      )ERASE FAC                     Erase function FAC
      FAC 5                          Function FAC no longer exists
SYNTAX ERROR
      FAC 5
      ∧
      ∇Z←BIN N                       An (erroneous) function for
[1]   LA:Z←(Z,0)+0,Z                      binomial coefficients
[2]   →LA×N≥ρZ∇
      BIN 3
VALUE ERROR
BIN[1] LA:Z←(Z,0)+0,Z                Suspended execution
                    ∧
      Z←1                            Assign value to Z
      →1                             Resume execution
1   3   3   1                        Binomial coefficients of order 3
      BIN 4
VALUE ERROR                          Same error (local variable Z
BIN[1] L1:Z←(Z,0)+0,Z                     does not retain its value)
                    ∧
      ∇BIN[∘1]Z←1∇                   Insert line to initialize Z
      )SI                            Display state indicator
BIN[1] *                             Suspended on line 1 of BIN
      →1                             Resume execution (BIN now correct)
1   4   6   4   1
      ∇BIN[⎕]∇                       Display revised function
    ∇ Z←BIN N                             and close definition
[1]   Z←1
[2]   LA:Z←(Z,0)+0,Z
[3]   →LA×N≥ρZ
    ∇
      SΔBIN←2                        Set stop on line 2
      Q←BIN 3                        Execute BIN

BIN[2]                               Stop due to stop control
      Z                              Display current value of Z
1
      →2                             Resume execution

BIN[2]                               Stop again on next iteration
      →2                             Resume

BIN[2]                               Stop again
      →0                             Branch to 0 (terminate)
```

INPUT AND OUTPUT

```
      ∇MULTDRILL N;Y;X                A multiplication drill
[1]    Y←?N                           ρN random integers
[2]    Y                              Print the random factors
[3]    X←⎕                            Keyboard input
[4]    →0×ιX='S'                      Stop if entry is the letter S
[5]    →ιX=×/Y                        Repeat if entry is correct product
[6]    'WRONG, TRY AGAIN'             Prints if preceding branch fails
[7]    →3∇                            Branch to 3 for retry
       MULTDRILL 12 12                Drill for pairs in range 1 to 12
2  10
⎕:                                    Indicates that keyboard entry
       37                                  is awaited
WRONG, TRY AGAIN
⎕:
       20
6  7
⎕:
       'S'                            Entry of letter S stops drill
       ∇Z←ENTERTEXT                   Example of character (⍞) input
[1]    Z←''                           Make Z an empty vector
[2]    D←ρZ                           D is the length of Z
[3]    Z←Z,⍞                          Append character keyboard entry
[4]    →2×D≠ρZ                        Branch to 2 if length increased
[5]    ∇                                   (i.e., entry was not empty)
       Q←ENTERTEXT
THIS IS ALL                           Keyboard
 CHARACTER INPUT                           entries
                                      Empty input to terminate
       Q                              Display Q
THIS IS ALL CHARACTER INPUT
       N←5
       'NOTE: ι';N;' IS ';ιN          Mixed output statement
NOTE:ι5 IS 1  2  3  4  5
```

RECTANGULAR ARRAYS

```
       P←2 3 5 7
       ρP                             Dimension of P
4
       T←'OH MY'                      Character vector
       ρT
5
       P,P                            Catenation
2  3  5  7  2  3  5  7
       T,T
OH MYOH MY
       T,P
DOMAIN ERROR                          Characters cannot be catenated
       T,P                                 with numbers
       ∧
```

```
      M←2 3ρ2 3 5 7 11 13           Reshape to produce a 2×3 matrix
      M                             Display of an array of rank >1
                                        is preceded by a blank line
2   3   5
7  11  13
      2 4ρT                         A 2×4 matrix of characters

OH M
YOH
      6ρM                           A matrix reshaped to a vector
2  3  5  7  11  13
      ,M                            Elements in row-major order
2  3  5  7  11  13
      P←,M
      P[3]                          Indexing (third element of P)
5
      P[1 3 5]                      A vector index
2  5  11
      P[ι3]                         The first three elements of P
2  3  5
      P[ρP]                         Last element of P
13
      M[1;2]                        Element in row 1, column 2 of M
3
      M[1;]                         Row 1 of M
2  3  5
      M[1 1;3 2]                    Rows 1 and 1, columns 3 2

 5  3
 5  3
      A←'ABCDEFGHIJKLMNOPQ'         The alphabet to Q
      A[M]                          A matrix index produces
                                        a matrix result
BCE
GKM
      A[M[1 1;3 2]]

EC
EC
      M[1;]←15 3 12                 Respecifying the first row of M
      M

 15   3  12
  7  11  13
```

```
        Q←3 1 5 2 4 6               A permutation vector
        P[Q]                        Permutation of P
5   2   11   3   7   13
        Q[Q]                        A new permutation
5   3   4   1   2   6
        P[3]                        Present index origin is 1
5
        )ORIGIN 0                   Set index origin to 0
WAS 1
        P[3]
7
        P[0 1 2]                    First three elements of P
2   3   5
        ⍳5                          Result of index generator
0   1   2   3   4                        begins at origin
        )ORIGIN 1
WAS 0
        ⍳5
1   2   3   4   5
```

FUNCTIONS ON ARRAYS

```
        V←?3⍴9                      Vector of 3 random integers (1-9)
        M←?3 3⍴9                    Random 3 by 3 matrix
        N←?3 3⍴9                    Random 3 by 3 matrix
        V
2   1   7
        M

   7   9   4
   5   8   1
   1   5   7
        N
   1   4   1
   4   7   6
   9   8   5
        M+N                         Sum (element-by-element)

    8   13    5
    9   15    7
   10   13   12
```

```
      M⌈N                          Maximum

  7  9  4
  5  8  6
  9  8  7
      M≤N                          Comparison

 0 0 0
 0 0 1
 1 1 0
      +/V                          Sum-reduction of V
10
      ×/V                          Product-reduction
14
      +/[1]M                       Sum over first coordinate of M
13  22  12                              (down columns)
      +/[2]M                       Sum over second coordinate of M
                                        (over rows)
20  14  13
      +/M                          Sum over last coordinate
20  14  13
      ⌈/M                          Maximum over last coordinate
9  8  7
      X←1.5
      +/(1 2○X)*2                  Sin squared plus Cos squared
1
      ○/1 2,X                      Sin Cos X
0.07067822453
      Y←○/0 2,X                    (1-(COS X)*2)*.5
      Y
0.9974949866
      Y=1○X                        An identity
1
      M+.×N                        Ordinary matrix (+.× inner)
                                        product
   79  123   81
   46   84   58
   84   95   66
      M+.≤N                        An inner product
  1  1  1
  1  1  1
  2  3  2
      M+.×V                        +.× inner product with vector
51  25  56                              right argument
```

```
      V
2  1  7
      V∘.×⍳5
```
Outer product (times)

```
  2   4   6   8  10
  1   2   3   4   5
  7  14  21  28  35
      V∘.≤⍳9
```
Outer product

```
0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 1 1 1
      V∘.×M
```
An outer product of rank 3

```
 14  18   8
 10  16   2
  2  10  14

  7   9   4
  5   8   1
  1   5   7

 49  63  28
 35  56   7
  7  35  49
```
A blank line between planes

MIXED FUNCTIONS

```
      Q←?10⍴5
      Q
1  4  3  4  5  4  2  1  4  2
      +/[1]Q∘.=⍳5
2  2  1  4  1

      2 1⍉M

  7  5  1
  9  8  5
  4  1  7
      ⍉M

  7  5  1
  9  8  5
  4  1  7
```
A random 10 element vector (range 1 to 5)

Ith element of result is number of occurences of the value I in Q

Ordinary transpose of *M*

Ordinary transpose of M (monadic)

```
      T←2 3 4ρι24                         An array of rank 3
      T

  1   2   3   4
  5   6   7   8
  9  10  11  12

 13  14  15  16
 17  18  19  20
 21  22  23  24

      3 1 2⍉T                              Transpose of T (dimension
                                                of result is 3  4  2)
  1  13
  2  14
  3  15
  4  16

  5  17
  6  18
  7  19
  8  20

  9  21
 10  22
 11  23
 12  24
      1 1⍉M                                Diagonal of M
7  8  7
      1 1 2⍉T                              Diagonal section in first
                                                two coordinates of T
  1   2   3   4
 17  18  19  20
      X←○(0,ι5)÷6
      )DIGITS 4                            Set number of output digits to 4
WAS 10
      ⍉1 2 3∘.○X

 0.000E0     1.000E0     0.000E0           Table of sines, cosines,and
 5.000E¯1    8.660E¯1    5.774E¯1               tangents in intervals
 8.660E¯1    5.000E¯1    1.732E0                of 30 degrees
 1.000E0     1.744E¯16   5.734E15
 8.660E¯1   ¯5.000E¯1   ¯1.732E0
 5.000E¯1   ¯8.660E¯1   ¯5.774E¯1
```

```
      Q
1  4  3  4  5  4  2  1  4  2
      3⌽Q
4  5  4  2  1  4  2  1  4  3
      ¯3⌽Q
1  4  2  1  4  3  4  5  4  2
      0 1 2⌽[1]M

  7  8  7
  5  5  4
  1  9  1
      ¯2⌽[2]M

  9  4  7
  8  1  5
  5  7  1
      1 2 3⌽M

  9  4  7
  1  5  8
  1  5  7
      ⌽Q
2  4  1  2  4  5  4  3  4  1

      ⌽[1]M

  1  5  7
  5  8  1
  7  9  4
      ⌽M

  4  9  7
  1  8  5
  7  5  1
```

3⌽Q: Rotate to left by 3 places

¯3⌽Q: Rotate to right by 3 places

0 1 2⌽[1]M: Rotate columns by different amounts

¯2⌽[2]M: Rotation of rows all by 2 to right

1 2 3⌽M: Rotation of rows

⌽Q: Reversal of Q

⌽[1]M: Reversal of M along first coordinate

⌽M: Reversal along last coordinate

```
      U←Q>4
      U
0  0  0  0  1  0  0  0  0  0
      U/Q                          Compression of Q by logical
5                                       vector U
      (~U)/Q                       Compression by not U
1  4  3  4  4  2  1  4  2
      +/U/Q
5
      1 0 1/[1]M                   Compression along first
                                        coordinate of M
   7  9  4
   1  5  7
      1 0 1/M                      Compression along last
                                        coordinate
   7  4
   5  1
   1  7
      (,M>5)/,M                    ,M is  7 9 4 5 8 1 1 5 7
7  9  8  7                         All elements of M which exceed 5
      V←1 0 1 0 1
      V\ι3                         Expansion of iota 3
1  0  2  0  3
      V\M                          Expansion of rows of M

   7  0  9  0  4
   5  0  8  0  1
   1  0  5  0  7
      V\'ABC'                      Expansion of literal vector
A B C                                   inserts spaces
      10⊥1 7 7 6                   Base 10 value of vector 1 7 7 6
1776
      8⊥1 7 7 6                    Base 8 value of 1 7 7 6
1022
      (4ρ10)⊤1776                  4 digit base 10 representation
1  7  7  6                              of number 1776
      (3ρ10)⊤1776                  3 digit base 10 representation
7  7  6                                 of 1776
      10 10⊤1776
7  6
      10⊤1776
6
      24 60 60⊥1 3 25              Mixed base value of 1 3 25
3805                                    (time radix)
      24 60 60⊤3805                Representation of number 3805
1  3  25                                in time radix
      2⊥1 0 1 1 0                  Base 2 value
22
```

```
        M
   7  9  4
   5  8  1
   1  5  7
        )ORIGIN 0
WAS 1
        M[2;0]                    Indexing of matrix in 0-origin.
1                                      Note relation to indexing of
        (,M)[(ρM)⊥2,0]                 ravel of M
1
        )ORIGIN 1                 Restore 1-origin
WAS 0
        P
2  3  5  7  11  13
        P⍳7                       Index of 7 in vector P
4                                 7 is 4th element of P
        P⍳6                       6 does not occur in P, hence
7                                      result is 1+ρP
        P⍳4 5 6 7
7  3  7  4
        Q←5 1 3 2 4               A permutation vector
        R←Q⍳⍳ρQ                   R is the permutation inverse to Q
        R
2  4  3  5  1
        Q[R]
1  2  3  4  5
        A←'ABCDEFGHIJKLMNOPQ'
        A←A,'RSTUVWXYZ'
        A                         A is the alphabet
ABCDEFGHIJKLMNOPQRSTUVWXYZ
        A⍳'C'                     Rank of letter C in alphabet is 3
3
        J←A⍳'CAT'
        J
3  1  20
A[J]
CAT
```

```
      M←3 5⍴'THREESHORTWORDS'          A matrix of characters
      M
THREE
SHORT
WORDS
      J←A⍳M                            Ranking of M produces a matrix
      J
 20   8  18   5   5
 19   8  15  18  20
 23  15  18   4  19
      A[J]                             Indexing by a matrix produces
                                            a matrix
THREE
SHORT
WORDS
      3?5                              Random choice of 3 out of 5
5  1  2                                     without replacement
      6?5
DOMAIN ERROR
      6?5
       ∧
      X←8?8                            A random permutation vector
      X
4  6  7  2  5  1  8  3
      ⍋X                               Grading of X
6  4  8  1  5  2  3  7
      X[⍋X]                            Arrange in ascending order
1  2  3  4  5  6  7  8
      X[⍒X]                            Arrange in descending order
8  7  6  5  4  3  2  1
      U←A∊'NOW IS THE TIME'            Membership
      '01'[1+U]
00001001100011100011001000
      U/A
EHIMNOSTW
      (⍳8)∊3 7 5
0  0  1  0  1  0  1  0
```

APL BIBLIOGRAPHY*

Berry, P. C., _APL\360 Primer,_ IBM Corporation 1968.

Berry, P. C., _APL\1130 Primer,_ IBM Corporation 1968.

Breed, L. M., and R. H. Lathwell, "The Implementation of APL\360," _ACM Symposium on Experimental Systems for Applied Mathematics_, Academic Press, 1968.

Falkoff, A. D., and K. E. Iverson, "The APL\360 Terminal System," _ACM Symposium on Experimental Systems for Applied Mathematics_, Academic Press, 1968.

Falkoff, A. D., K. E. Iverson, and E. H. Sussenguth, "A Formal Description of System/360," _IBM Systems Journal_, Volume 3, Number 3, 1964.

Iverson, K E., _A Programming Language,_ Wiley, 1962.

Iverson, K. E., _Elementary Functions: an algorithmic treatment,_ Science Research Associates, 1966.

Iverson, K. E., "The Role of Computers in Teaching," _Queen's Papers in Pure and Applied Mathematics_, Volume 13, Queen's University, Kingston, Canada, 1968.

Lathwell, R. H., _APL\360: Operator's Manual_, IBM Corporation, 1968.

Lathwell, R. H., _APL\360: System Generation and Library Maintenance,_ IBM Corporation, 1968

Pakin, S., _APL\360 Reference Manual_, Science Research Associates, 1967.

Rose, A. J., Videotaped APL Course, IBM Corporation 1968.

Smillie, K. W., _Statpack 1: An APL Statistical Package_, Publication No. 9, Department of Computing Science, University of Edmonton, Canada, 1968.

* For additional information or assistance regarding APL, see your local IBM representative.