

*NB. J BY EXAMPLE*

*NB. J is product of JSoftware Inc. <http://jsoftware.com>  
 NB. v3 07/04/2005 (C) Oleg Kobchenko <http://olegykj.sourceforge.net>*

*NB. simple arithmetic =====*

```

  2 + 2      NB. comment is 'NB.'
4
  2 - 3      NB. negative numbers use '_'
_1
  2 * 3 + 4  NB. no precedence, right to left
14
  (2 * 3) + 4 NB. parentheses changes order
10
  3 % 4      NB. division represented by '%'
0.75
  *: 4      NB. square
16
  %: 4      NB. square root
2
  % 4      NB. 1/x
0.25

```

*NB. operations using lists =====*

```

  2 * 1 2 3  NB. numeric list with space separators
2 4 6
  1 2 3 % 2 4 6 NB. list to list operations, same size
0.5 0.5 0.5
  #1 2 3     NB. size of vector
3
  3$1       NB. generate sequence of same numbers
1 1 1
  5$1 2     NB. or from a list of given elements
1 2 1 2 1

```

*NB. list elements =====*

```

  {.1 2 3   NB. first element
1
  {:1 2 3   NB. last element
3
  }.1 2 3   NB. rest without first element
2 3
  }:1 2 3   NB. rest without last element
1 2
  |.1 2 3   NB. reverse
3 2 1

```

*NB. indexing and sorting =====*

```

  1{1 2 3   NB. indexing is zero-based
2
  1 0{1 2 3 NB. index can be vector too
2 1
  i.3       NB. generate zero-based sequence
0 1 2
  2 4 6 i. 4 NB. index of given element(s)
1
  /:2 1 6   NB. indices of sorted order
1 0 2
  /:~2 1 6  NB. sort vector
1 2 6      NB. F~y <=> y F y

```

*NB. list aggregation =====*

```

1 2 3,10 20      NB. join vectors
1 2 3 10 20
  1 + 2 + 3      NB. sum of elements
6
+ / 1 2 3        NB. insert '+' between elements
6
+ / \ 1 2 3      NB. running sum of elements
1 3 6
  1, (1+2), (1+2+3) NB. same as this
1 3 6
  2+ / \ 1 2 3 4 5 NB. sum or running pairs
3 5 7 9
  _2+ / \ 1 2 3 4 5 NB. non-intersecting pairs
3 7 5
  (< 1 2), 3 4 6; 7 6 NB. < is boxing, ; is box and join
+-----+
| 1 2 | 3 4 6 | 7 6 |
+-----+
  > { . 3 4 6; 7 6   NB. > is unboxing
3 4 6

```

*NB. function combinations =====*

```

(+ *: ) 4      NB. hook (F G) y <=> y F (G y)
20      NB. a + a^2
(%, , *: ) 4   NB. fork (F G H) y <=> (F y) G (H y)
2 16      NB. [sqrt(a), a^2]
*:@(+/) 2 3    NB. composition (F o G) y <=> F G y
25      NB. (a + b)^2
2 +&*: 3       NB. x F & G y <=> (G x) F (G y)
13      NB. a^2 + b^2
2 (+&*: + 2: * *) 3 NB. (a + b)^2 = a^2 + b^2 + 2ab
25      NB. 0: 1: 2: ... are const functions
3 +&.*: 4      NB. F&.G y <=> (G^:_1) F G y
5      NB. sqrt(a^2 + b^2)

```

*NB. user defined functions and arguments =====*

```

m1=: -          NB. ambivalent tacit
m2=: 3 : '-y.'  NB. monadic explicit
m3=: 4 : 'x.-y.' NB. dyadic explicit

(m1 , m2 , 0&m3) 4 NB. monadic use, 0& is bonding
_4 _4 _4
  3 (m1 , (+ m2) , m3) 4 NB. dyadic use, hook for dyadization
_1 _1 _1
  (m1 , m3) / 3 4      NB. distribute arguments: dyadization
_1 _1
  3 (m1 , m4) @ , 4    NB. collect arguments: monadization
_3 _4 _3 _4

```

*NB. exponent and logarithm =====*

```

1x1 2x1 1x2      NB. e, 2e, e squared
2.71828 5.43656 7.38906
  ^2              NB. exponent, e^2
7.38906
  2^16            NB. exponent base 2, 2^16
65536
  ^. 1x2          NB. logarithm, ln e^2
2
  2^.65536        NB. logarithm base 2, log2 65536
16

```

*NB. trigonometry* =====

```

1p1 2p1 1p2          NB. pi, 2 pi, pi squared
3.14159 6.28319 9.8696
load 'trig'          NB. load trigonometry library
cos 1p1              NB. cosine of pi
_1
(*:cos 1p1) + *:sin 1p1 NB. theorem of trigonometry
1
(cos +&*: sin) 1 2p1 1p2 NB. same using fork and &
1 1 1

```

*NB. matrices* =====

```

1 2 3 */ 1 2 3 NB. outer product: multiplication table
1 2 3          NB. same as */~ 1 2 3
2 4 6
3 6 9
=~/i.3        NB. identity matrix, also =@i. (self-classify)
1 0 0          NB. F~y <=> y F y
0 1 0
0 0 1
]M=. i.2 3     NB. generate matrix
0 1 2
3 4 5
2 2$0 1 1 1   NB. reshape given vector to matrix
0 1
1 1

```

*NB. structural transforms* =====

```

,N=: i.2 2 3      NB. ravel: list of atoms
0 1 2 3 4 5 6 7 8 9 10 11
,"2 N            NB. ravel each sub-matrix
0 1 2 3 4 5
6 7 8 9 10 11

( |; |:; |. |."1;l&|. ) M=. 3 3$'ABC123!@#' NB. character matrix
+---+---+---+---+---+   NB. ] returns argument
|ABC|A1!|!@#|CBA|123|   NB. |: transposes
|123|B2@|123|321|!@#|   NB. |. reverses outer list
|!@#|C3#|ABC|#@!|ABC|   NB. |."1 reverses inner list
+---+---+---+---+---+   NB. 1|. rotates outer list

;:^:_1 </.M        NB. oblique: secondary diagonals
A B1 C2! 3@ #     NB. same as (</.~&, +"0/~@i.@#) M
NB. ;:^:_1 is inverse of boxing tokens

i.@# } M          NB. main diagonal
A2#

```

*NB. selection* =====

```

1{1{1{N          NB. repetitive selection of items From list
10
1{^:3 N          NB. apply select 3 times
10
(<1 1 1){N       NB. scatter select
10
1 1 1 ({~ <~)~ N NB. using unboxed list
10

```

*NB. factorial and binomial =====*

```

! 1+i.5      NB. factorial
1 2 6 24 120
*/\ 1+i.5    NB. running product
1 2 6 24 120
!/~ i.5      NB. binomial coefficients
1 1 1 1 1
0 1 2 3 4
0 0 1 3 6
0 0 0 1 4
0 0 0 0 1
+/@(! |.)\i. 15 NB. fibonacci: sum of second diagonal of binomial matrix
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610

```

*NB. dot product =====*

```

1 2 3(+/. *)1 2 3      NB. dot product
14
M=: 2 2$0 1 1 1      NB. assignment
dot=: +/. *           NB. expression given a name
dot~ M               NB. matrix squared
1 1
1 2
dot^(15)~ M          NB. matrix to the power of 15, also fibonacci
610 987
987 1597
{:@{"2 dot^:(<15)~ M NB. F^n is apply F n times accumulatively
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610

```

*NB. randomness and probability =====*

```

]A=: 5 ?@$ 0          NB. 5 random floats from 0..1
0.57708 0.542732 0.488337 0.26004 0.0101683
]B=: 10 ?@$ 2         NB. coin toss
1 1 1 0 0 1 1 0 1 0
]C=: 3 ? 3           NB. deal 3 out of 3 cards in certain order
1 2 0
(<./ , >./) A        NB. min and max over the list
0.0101683 0.57708
B i. 0               NB. first zero
3
(+/. % #) C-:"1 (?~"0) 10000#3 NB. method monte carlo
0.1637              NB. -: is list equality, F^n is rank modifier
%!3                 NB. exact probability of 3 cards in given order
0.166667

```

*NB. unique elements =====*

```

]D=.~. S=. 'mississippi' NB. distinct (nub)
misp
]K=. D i. S           NB. key (index)
0 1 2 2 1 2 2 1 3 3 1
K </. S              NB. group by key
+-----+-----+
|m|iiii|ssss|pp|
+-----+-----+
K #/. S              NB. frequencies
1 4 4 2

]I=. ~.: S           NB. sieve of nub
1 1 1 0 0 0 0 1 0 0 NB. where D is in S

I # S               NB. filter by sieve to get D
misp
+/"1 =S            NB. = is classify, bool matrix of S vs D
1 4 4 2           NB. where items of D are in S

```