```
SYM    Construct symbolic numbers, variables and objects.
   S = SYM(A) constructs an object S, of class 'sym', from A.
   If the input argument is a string, the result is a symbolic number
   or variable.  If the input argument is a numeric scalar or matrix,
   the result is a symbolic representation of the given numeric values.

   x = sym('x') creates the symbolic variable with name 'x' and stores the
   result in x.  x = sym('x','real') also assumes that x is real, so that
   conj(x) is equal to x.  alpha = sym('alpha') and r = sym('Rho','real')
   are other examples.  Similarly, k = sym('k','positive') makes k a
   positive (real) variable.  x = sym('x','unreal') makes x a purely
   formal variable with no additional properties (i.e., insures that x
   is NEITHER real NOR positive).
```

```
 SYMS   Short-cut for constructing symbolic objects.
   SYMS arg1 arg2 ...
   is short-hand notation for
      arg1 = sym('arg1');
      arg2 = sym('arg2'); ...
```

```
>> syms a s t w x
>> whos
  Name      Size                   Bytes  Class

  a         1x1                      126  sym object
  ans       1x1                      130  sym object
  s         1x1                      126  sym object
  t         1x1                      126  sym object
  w         1x1                      126  sym object
  x         1x1                      126  sym object

Grand total is 14 elements using 760 bytes
```

```
LAPLACE Laplace transform.
   L = LAPLACE(F) is the Laplace transform of the scalar sym F with
   default independent variable t.  The default return is a function
   of s.  If F = F(s), then LAPLACE returns a function of t:  L = L(t).
   By definition L(s) = int(F(t)*exp(-s*t),0,inf), where integration
   occurs with respect to t.

   L = LAPLACE(F,t) makes L a function of t instead of the default s:
   LAPLACE(F,t) <=> L(t) = int(F(x)*exp(-t*x),0,inf).

   L = LAPLACE(F,w,z) makes L a function of z instead of the
   default s (integration with respect to w).
   LAPLACE(F,w,z) <=> L(z) = int(F(w)*exp(-z*w),0,inf).

   Examples:
     syms a s t w x
     laplace(t^5)           returns   120/s^6
     laplace(exp(a*s))      returns   1/(t-a)
     laplace(sin(w*x),t)    returns   w/(t^2+w^2)
     laplace(cos(x*w),w,t)  returns   t/(t^2+x^2)
     laplace(x^sym(3/2),t)  returns   3/4*pi^(1/2)/t^(5/2)
     laplace(diff(sym('F(t)')))   returns   laplace(F(t),t,s)*s-F(0)
```

```
>> syms a s t w x
>> laplace(t^5)

ans =

120/s^6

>> pretty(ans)

                              120
                              ---
                               6
                              s
```

```
ILAPLACE Inverse Laplace transform.
   F = ILAPLACE(L) is the inverse Laplace transform of the scalar sym L
   with default independent variable s.  The default return is a
   function of t.  If L = L(t), then ILAPLACE returns a function of x:
   F = F(x).
   By definition, F(t) = int(L(s)*exp(s*t),s,c-i*inf,c+i*inf)
   where c is a real number selected so that all singularities
   of L(s) are to the left of the line s = c, i = sqrt(-1), and
   the integration is taken with respect to s.

   F = ILAPLACE(L,y) makes F a function of y instead of the default t:
      ILAPLACE(L,y) <=> F(y) = int(L(y)*exp(s*y),s,c-i*inf,c+i*inf).
   Here y is a scalar sym.

   F = ILAPLACE(L,y,x) makes F a function of x instead of the default t:
   ILAPLACE(L,y,x) <=> F(y) = int(L(y)*exp(x*y),y,c-i*inf,c+i*inf),
   integration is taken with respect to y.

   Examples:
     syms s t w x y
     ilaplace(1/(s-1))              returns   exp(t)
     ilaplace(1/(t^2+1))            returns   sin(x)
     ilaplace(t^(-sym(5/2)),x)      returns   4/3/pi^(1/2)*x^(3/2)
     ilaplace(y/(y^2 + w^2),y,x)    returns   cos(w*x)
     ilaplace(sym('laplace(F(x),x,s)'),s,x)   returns   F(x)
```

```
>> ilaplace(1/(s-1))

ans =

exp(t)
```