

一、运算符

1.1 算术运算符

【例】若 $x=2$ ，求解 $3x^{10} + 6/(15-\sqrt{x})$ 的数值。

```
x = 2;
y = 3*x^10 + 6/(15-x^(1/2));
```

1.2 关系运算符

① 关系运算符往往用在约束条件中，用来指定约束条件左右两边必须满足的关系。

② Lingo 只有三种关系运算符：“=”、“>=”以及“<=”。

没有单独的“>”和“<”，若出现，Lingo 则视为省略了“=”。

若想严格表达 A 大于 B，可以用以下方式：

```
B = 10;
e = 0.0001;
A - e > B;
```

1.3 逻辑运算符

① 定义：

分类	运算符	理解	作用
两个数字之间	# eq #	equal	两个运算对象相等为真
	# ne #	not equal	两个运算对象不相等为真
	# gt #	greater than	左边大于右边为真
	# ge #	greater equal	左边大于等于右边为真
	# lt #	less than	左边小于右边为真
	# le #	less equal	左边小于等于右边为真
两个逻辑表达式之间	# not #	非门	单目运算符，表示取反
	# and #	与门	左右两边均正确才为真
	# or #	或门	左右两边均错误才为假

② 逻辑运算符唯一出现的位置：

- a) for 循环与 sum 求和。
- b) if 判断中。

【例 1】若矩阵 $a=[6,5,4,3,2,1]$ ，用集合的语言求解 $a(5)+a(6)$ 。

```
model:
sets:
factory / 1..6 / : a;
endsets
data:
a = 6,5,4,3,2,1;
enddata
y = @sum( factory(i) | i#ge#5 : a(i) );
end
```

上式中， $i\#ge\#5$ 对 sum 求和的范围进行了限制，当然也可以写 $i\#gt\#4$ 。

【例 2】若矩阵 a 由六个元素组成，且 $a(i) > 5$, $i=1,2,5,6$ 。求矩阵 a 各元素求和的最小值。

```
model:
sets:
factory / 1..6 / : a;
endsets
min = @sum(factory : a);
@for( factory(i) | i#lt#3 #or# i#ge#5 : a(i) > 5 );
end
```

【例 3】若矩阵 $a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ ，求矩阵上三角之和（含主对角线）。

```
model:
sets:
fac / 1..3 / : ;
coo(fac,fac) : a;
endsets
data:
a = 1,2,3
    4,5,6
    7,8,9;
enddata
y = @sum( coo(i,j) | i#le#j : a(i,j) );
end
```

【注】以上三个问题只是为了熟练逻辑运算符在 for 与 sum 中的使用。

二、Lingo 内置函数

2.1 if 判断

【例 1】 用 Lingo 表达出分段函数 $y = \begin{cases} x+10, & x \geq 0 \\ x-10, & x < 0 \end{cases}$ ，并求出 x 为一系列数值时的结果。

【解】

- ① Lingo 默认所有变量不为负数，故应先进行定义域自由化。
- ② if 函数语法简单，看一眼即可学会。

```
@free(x);
@free(y);
x = -10;      ! 给 x 一个随机的初值
y = @if( x#ge#0 , x+10 , x-10 );
```

【例 2】 用 Lingo 表达出分段函数 $y = \begin{cases} 4x, & 0 \leq x \leq 500 \\ 500 + 3x, & 500 < x \leq 1000 \\ 1500 + 2x, & x > 1000 \end{cases}$ 。

【解】

- ① $0 \leq x \leq 500$ 可以 $x\#ge\#0 \#and\# x\#le\#500$ ，也可以直接 $x\#le\#500$ 。
- ② if 函数的嵌套功能：

```
x = 1500;      ! 给 x 一个随机的初值
y = @if( x#le#500 , 4*x , @if( x#gt#1000 , 1500+2*x , 500+3*x ));
```

【提示】

- ① if 函数通常仅仅在分段函数处出现，一般其出现频率、使用次数十分之低。
- ② Lingo 中的 if 函数，必须自带一个 else。

2.2 变量定界函数

函数	作用	记忆
<code>@bin(x)</code>	限制 x 只能取 0 或 1，0-1 规划中特别有用	有病就是 1，没病就是 0
<code>@gin(x)</code>	限制 x 为整数，在整数规划中特别有用	gin 谐音哽塞的哽，哽成整数
<code>@bnd(a,x,b)</code>	限制 $a \leq x \leq b$ ，推荐直接替换两个约束条件	b 和 d 对称，两边夹住变量 n
<code>@free(x)</code>	取消对变量 x 非负的限制，使其定义域自由	老朋友，不用记了

【@free】求函数 $z = (x+2)^2 + (y-2)^2$ 的最小值。

【解】

```
@free(x);
@free(y);
min = (x+2)^2 + (y-2)^2;
```

【注】0.5473e-25 是科学计数法，意为 0.5473×10^{-25} 。此语句在 matlab 同为科学计数法。

【@bnd】求函数 $y = 2x$ 在 (1,3) 之间的最大值。

【解】

```
@bnd(1,x,3);
max = 2*x;
```

【注】第一句可写为 $x > 1; x < 3$; 替代，但不论从速度还是约束条件数量，都不如用 @bnd 函数。

【@bin】已知 $a = [2, 9, 3, 8, 10, 6, 4, 10]$ 以及 $b = [1, 3, 4, 3, 3, 1, 5, 10]$ ，求以下线性规划：

$$\begin{aligned} \max \quad & z = \sum_{i=1}^8 a_i x_i \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^8 b_i x_i \leq 15 \\ x_i = 1 \text{或} 0, \quad i = 1, 2, \dots, 8 \end{cases} \end{aligned}$$

【解】

```
model:
```

```
sets:
```

```
factory / 1..8 / : a,b,x;
```

```
endsets
```

```
data:
```

```
a = 2,9,3,8,10,6,4,10;
```

```
b = 1,3,4,3,3,1,5,10;
```

```
enddata
```

```
max = @sum(factory : a*x );
```

```
@sum(factory : b*x ) < 15;
```

```
@for( factory : @bin(x) );
```

```
end
```

【@gin】已知 $a = [2.1 \ 1.0 \ 1.8 \ 1.2 \ 2.0 \ 1.2]$ 以及 $b = [6 \ 125 \ 12500 \ 345 \ 5]$ ，求整数规划：

$$\begin{aligned} \max \quad & z = \sum_{i=1}^6 a_i x_i \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^6 c_{ij} x_i \leq b_j, \quad j = 1, 2, \dots, 5 \\ \sum_{i=1}^6 x_i = 14 \\ x_2 \leq 3, x_4 \leq 2 \\ 1 \leq x_i \leq 4, \quad i = 1, 3, 5, 6 \end{cases} \end{aligned} \quad c = \begin{bmatrix} 0.45 & 20 & 415 & 22 & 0.3 \\ 0.45 & 28 & 4065 & 5 & 0.35 \\ 0.65 & 40 & 850 & 43 & 0.6 \\ 0.4 & 25 & 75 & 27 & 0.2 \\ 0.5 & 26 & 76 & 48 & 0.4 \\ 0.5 & 75 & 235 & 8 & 0.6 \end{bmatrix}$$

【解】

```
model:
sets:
    factory / 1..6 / : a,x;
    plant / 1..5 / :b;
    coo(factory,plant) : c;
endsets
data:
    a=2.1 1.0 1.8 1.2 2.0 1.2;
    b=6 125 12500 345 5;
    c=0.45 20 415 22 0.3
        0.45 28 4065 5 0.35
        0.65 40 850 43 0.6
        0.4 25 75 27 0.2
        0.5 26 76 48 0.4
        0.5 75 235 8 0.6;
enddata
max = @sum( factory : a*x );
@for( plant(j) : @sum(factory(i):c(i,j)*x(i)) <= b(j) );
@sum( factory : x )=14;
x(2)<=3;
x(4)<=2;
@for( factory(i) | i#ne#2 #and# i#ne#4 : @bnd(1,x(i),4) );
@for( factory : @gin(x) );
end
```

2.3 数学函数

类别	函数名	返回值
三角函数	<code>@sin(x)</code>	返回 x 的正弦值
	<code>@cos(x)</code>	返回 x 的余弦值
	<code>@tan(x)</code>	返回 x 的正切值
指数对数	<code>@log(x)</code>	返回 x 的自然对数值，其他底数用换底公式
	<code>@exp(x)</code>	返回 e^x 的值，因 e 的数值无法敲入而诞生
其它	<code>@abs(x)</code>	返回 x 的绝对值
	<code>@sigh(x)</code>	返回 x 的符号值， $x \geq 0$ 为 1， $x < 0$ 为 -1
	<code>@floor(x)</code>	返回 x 的整数部分，向靠近 0 的方向取整
比较大小	<code>@smax(x1, x2, ..., xn)</code>	返回其中的最大值
	<code>@smin(x1, x2, ..., xn)</code>	返回其中的最小值

【例 1】求解 $\sin(3.14159) + \log_2^{1024} + |-10| + e^0$ 。

【解】

```
y = @sin(3.14159) + @log(1024) / @log(2) + @abs(-10) + @exp(0);
```

2.4 集合操作函数

设 factoty 工厂生产 6 个元素的矩阵：

类别	函数名	作用
熟人	@for(factory : a>0)	循环
	@sum(factory : a)	求和
有用	@prod(factory : a)	求积
	@max(factory : a)	求最大值
	@min(factory : a)	求最小值
鸡肋	@in(factory , c)	判断常数 c 是否在集合中
	@size(factory)	返回工厂可生产矩阵的长度

【举例】

```
model:
sets:
factory / 1..6 / : a;
endsets
data:
a = 6,5,4,3,2,1;
enddata
prod = @prod(factory : a);
greater = @max(factory : a);
less = @min(factory : a);
in = @in(factory,5);
size = @size(factory);
end
```