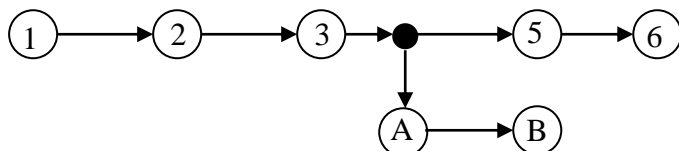


## AutoLisp 簡介

- 本簡介僅列出較常用之基本觀念。詳細且完整之功能，請自行參閱當時最新版本之 AutoLisp，或 <http://home.netcom.com/~rogh/autolisp/>。
- AutoLisp 乃是依 Lisp 程式語言為基礎，加以擴充，使之可呼叫 AutoCAD 繪圖功能。
- Lisp = List Programming
- List (串列)：一種資料結構



- 表示法：(1 2 3 (A B) 5 6)
- atom：最基本之構成原子。如：1, 2, 3, A, B, 5, 6
- element：構成 list 的元素，可以是 atom(如：1, 2, 3, 5, 6)或者子串列(sub-list, 如：(A B))
- Lisp 的特性
  - 將 list 視為函式(function)，其中，第一個元素為運算子(operator)，餘為運算元(operand)。例如：一般數學式  $3 + 2$ ，Lisp 式為 `(+ 3 2)`； $f(x, y)$  的 Lisp 式為 `(f x y)`。
  - 資料(data)為一種特殊的函式，其運算子為 quote。例如：資料(a,b,c)的 Lisp 式為 `(quote a b c)` 或簡示為 `'(a b c)`
  - 函式內部指述(statement)，依序(sequential)執行。
  - 函式之間無必然前後之執行關係(non-procedural)。
  - 不分大小寫，系統一律回應以大寫。
- 資料型態
  1. integer
  2. real
  3. string 字串，以 " " 括之，其中可含下列特殊字元：

特殊字串	意義(對應鍵盤)
\\	\
\e	Escape 鍵
\n	New line
\r	Return 鍵
\t	Tab 鍵
\nnn	八進位碼為 nnn 之字元
- 常數

1. T : true
  2. Nil : false
- ; 註解符號，置於行首。該行不執行。
  - (Auto)Lisp 為交談式(interactive)的操作環境，使用 interpreter (而非 compiler)。可以在螢幕上交談式執行，或寫成程式，以批次式執行(詳後)。
  - 常用指令集(完整指令集，請自行參考 AutoLisp 相關手冊)

#### 數學運算指令

運算子	運算元	說明
+	N1 N2 N3 ...	$N1 + N2 + N3 + \dots$
-	N1 N2 N3 ...	$N1 - N2 - N3 - \dots$
*	N1 N2 N3 ...	$N1 * N2 * N3 * \dots$
/	N1 N2 N3 ...	$N1 / N2 / N3 / \dots$
=	N1 N2 N3 ...	如果 $N1=N2=N3=\dots$ ，傳回 t，否則 nil
/=	N1 N2	如果 N1 不等於 N2，傳回 t，否則 nil
<	N1 N2 N3 ...	如果 $N1 < N2 < N3 < \dots$ ，傳回 t，否則 nil
<=	N1 N2 N3 ...	如果 $N1 \leq N2 \leq N3 \leq \dots$ ，傳回 t，否則 nil
>	N1 N2 N3 ...	如果 $N1 > N2 > N3 > \dots$ ，傳回 t，否則 nil
>=	N1 N2 N3 ...	如果 $N1 \geq N2 \geq N3 \geq \dots$ ，傳回 t，否則 nil
1+	N	$N+1$
1-	N	$N-1$
Abs	N	N 的絕對值
Rem	N1 N2	$N1/N2$ 的餘數
Sin	$\theta$ (弧度量[ $\pi$ ])	$\sin(\theta)$
Cos	$\theta$ (弧度量[ $\pi$ ])	$\cos(\theta)$
Atan	$\theta_1 \theta_2$ (弧度量[ $\pi$ ])	$\tan^{-1}(\theta_1/\theta_2)$ , $\theta_2 \neq 0$
Exp	N	$e^N$
Expt	Base power	$(\text{Base})^{\text{power}}$
Log	N	$\log_e(N)$ , 自然對數值
GCD	N1 N2 N3 ...	N1 N2 N3 ... 的最大公約數
Max	N1 N2 N3 ...	取最大值
Min	N1 N2 N3 ...	取最小值
Sqrt	N	N 的正平方根

#### 串列運算

運算子	運算元	說明
quote	exp	回應 exp 本身，亦可簡示為 'exp 例：(quote a) 或 'a (quote (a b c)) 或 '(a b c)
setq	sym1 exp1 sym2 exp2 sym3 exp3 ....	將 $\text{exp}_i$ 值指定給 $\text{sym}_i$ 例：(setq a 5.0) (setq s "it") (setq x '(a b))
list	Exp1 Exp2 ...	以 Exp1 Exp2 ... 為元素，合成一串列。例：

		$(\text{list } 'a 'b 'c) \rightarrow (a \ b \ c)$ $(\text{list } 'a '(b \ c) 'd) \rightarrow (a \ (b \ c) \ d)$ $(\text{list } 3.9 \ 6.7) \rightarrow (3.900000 \ 6.700000)$
cons	Element list	將 element 併入 list 前端。例： $(\text{cons } 'a '(b \ c \ d)) \rightarrow (a \ b \ c \ d)$
append	Exp1 Exp2 Exp3 ....	將 Exp1 Exp2 Exp3 ....合成一個 list。例： $(\text{append } '(a \ b) '(c \ d)) \rightarrow (a \ b \ c \ d)$
car	list	取 list 的第一個元素。若為空 list，傳回 nil。例： $(\text{car } '(a \ b \ c)) \rightarrow a$ $(\text{car } '((a \ b) \ c)) \rightarrow (a \ b)$ $(\text{car } '()) \rightarrow \text{nil}$
cdr	list	去除第一個元素後，剩下的 list。若原 list 為空串列，傳回 nil。例： $(\text{cdr } '(a \ b \ c)) \rightarrow (b \ c)$ $(\text{cdr } '((a \ b) \ c)) \rightarrow (c)$ $(\text{cdr } '()) \rightarrow \text{nil}$
Cadr, caddr, Cddr,...等	list	$(\text{Cadr } \text{list}) = (\text{car } (\text{cdr } \text{list}))$ $(\text{Caddr } ...) = (\text{car } (\text{cdr } (\text{cdr } \text{list})))$ $(\text{Cddr } ...) = (\text{cdr } (\text{cdr } \text{list}))$ ，依此類推。例： $(\text{Cddr } '(a \ b \ c \ d)) \rightarrow '(c \ d)$ $(\text{Cadr } '(a \ b \ c \ d)) \rightarrow b$
nth	N list	傳回 list 第 N 個元素，(N=0 為第一個元素)。例： $(\text{nth } 3 '(a \ b \ c \ d \ e)) \rightarrow d$ $(\text{nth } 5 '(a \ b \ c \ d \ e)) \rightarrow \text{nil}$ $(\text{nth } 0 '(a \ b \ c \ d \ e)) \rightarrow a$
last	list	list 的最後一個元素。例： $(\text{last } '(a \ b \ c \ d)) \rightarrow d$
length	list	串列 list 的元素個數。例： $(\text{length } '(a \ b \ c \ d)) \rightarrow 4$
atom	item	若 item 為元素，傳回 t，否則 nil
eval	exp	傳回 exp 之值。例： $(\text{eval } (+ \ 3 \ 5)) \rightarrow 8$ $(\text{setq } a \ 123)$ $(\text{setq } b \ 'a)$ $(\text{eval } 4.0) \rightarrow 4.000000$ $(\text{eval } (\text{abs } -10)) \rightarrow 10$ $(\text{eval } a) \rightarrow 123$ $(\text{eval } b) \rightarrow 123$
reverse	list	串列之反順序。例： $(\text{reverse } '((a) \ b \ c)) \rightarrow (c \ b \ (a))$
assoc	item alist	找 alist 中之某一 item。若找不到，傳回 nil。例： $(\text{setq } al \ '((name \ box) \ (width \ 3) \ (size \ 4.7)))$ $(\text{assoc } 'size \ al) \rightarrow 4.7$ $(\text{assoc } 'weight \ al) \rightarrow \text{nil}$

#### 資料型態轉換

運算子	運算元	說明
chr	number	將 ASCII 碼 number 轉成字元。例：

		(chr 65) → “A” (chr 66) → “B”
ascii	string	字串 string 之第一個字元的 ASCII 碼。例： (ascii “A”) → 65 (ascii “BIG”) → 66
atof	string	數字字串，轉成實數。例： (atof “97.1”) → 97.100000 (atof “3”) → 3.000000
atoi	string	數字字串，捨去小數，轉成整數。例： (atoi “97”) → 97 (atoi “97.3”) → 97 (atoi “97.9”) → 97
itoa	int	將整數 int 轉成字串。例： (itoa 33) → “33” (itoa -17) → “-17”
float	number	將數 number 轉成實數。例： (float 3) → 3.000000 (float 3.7) → 3.700000
fix	number	刪除小數。例 (fix 3.7) → 3.000000

#### 控制指令

運算式	說明
(if <i>testexp</i> <i>thenexp</i> <i>elseexp</i> )	<ul style="list-style-type: none"> <li>● else 部份可省略。</li> <li>● 當 <i>testexp</i> 為偽，且無 <i>elseexp</i> 時，傳回 nil</li> </ul> 例： (if (= 1 2) “YES!” “NO!”) → “NO!” (if (= 2 (+ 1 1)) “YES”) → “YES” (if (= 2 (+ 3 4)) “YES”) → nil
(cond ( <i>test1 result1</i> ) ( <i>test2 result2</i> ) ..... )	<ul style="list-style-type: none"> <li>● 設 <i>testi</i> 為依序第一個 t 的 <i>testexp</i>，傳回 <i>resulti</i> 之結果</li> </ul> 例： (cond ((= 3 a) (1+ a)) (t -10)) 如果 原已 (setq a 3) → 4 否則 → -10
(foreach name list exp1 exp2 ...)	<ul style="list-style-type: none"> <li>● 把 list 的元素，分別指定給 name，並執行 exp1, exp2, ... (name 為 exp 的一個變數)</li> </ul> 例： (foreach n ‘(a b c) (print n)) 等於執行 (print a) (print b) (print c)
(repeat N exp1 exp2 ...)	<ul style="list-style-type: none"> <li>● 重複執行 N 次 exp1, exp2, ...</li> </ul> 例： (setq a 10 b 100) (repeat 4 (setq a (+ a 10)))

	<pre>(setq b (+ b 10))</pre> <p>➔ 140 (結束時，a 為 50, b 為 140)</p> <ul style="list-style-type: none"> <li>● 本函式包含多個子函式。以最後子函式之傳回值，做為本函式之傳回值</li> </ul>
--	---

### 函式定義指令

函式	說明
(defun funcname (arguments) exp1 exp2 exp3 ...)	定義函式 funcname; 其引數為 arguments, 執行指令依序為 exp1, exp2, ...
(command autocadcommands)	執行 AutoCAD 指令

例：

- 以文書處理軟體，撰寫下列程式，並以「純文字」模式，在 D:\ 下，存成 AAA.LSP 檔。(檔名 AAA 可以任取，副檔名須為 LSP)

(DEFUN C:\aline( )

    (setq pt1 '(1.45 3.23))

    (setq pt2 (getpoint "Enter a point: "))

    (command "LINE" pt1 pt2 ""))

)

- 在進入 AutoCAD 環境後，  
指令：(load "D:\\AAA)  
指令：aline  
此時，AutoCAD 會把 aline 視同一個新的 AutoCAD 指令
- 同一個檔內可以定義多個指令。
- 如未在程式內加上 C:\，則必須在 AutoCAD 的指令行，下 (aline)，才會執行。
- 可以先在 AutoCAD 環境試作所有操作程序，記下各步驟之指令及引數，再寫成 (command ...)指令。

### AutoCAD/幾何函式

(getangle pt prompt)	原點與 pt 所成直線，與 X 軸夾角
(getdist pt prompt)	原點與 pt 點之距離
(getpoint pt prompt)	以滑鼠定點，傳回坐標值。Pt 乃定位用之參考點，非儲存傳回值之變數。
(getvar varname)	給 varname 變數值
(getint prompt)	輸入一個整數值
(getreal prompt)	輸入一個實數值
(getstring t prompt)	若有 t，輸入之字串可含空白
(angle pt1 pt2)	(pt1, pt2)所成線，與 X 軸之夾角
(distance pt1 pt2)	pt1 和 pt2 兩點距離

(polar pt angle d)	與 pt 點水平夾角 angle(弧度量)距離 d 之點
(inters pt1 pt2 pt3 pt4)	線段(pt1,pt2) 和 (pt3,pt4)之交點

#### 檔案處理函式

函式	說明
(open filename mode)	<ul style="list-style-type: none"> <li>● Mode : r : read w : write</li> <li>● Filename 可包含路徑名稱</li> </ul>
(read-char fileid)	
(read-line fileid)	
(write-char fileid)	
(write-line fileid)	
(close fileid)	關閉檔，並回應 nil

例：[注意其與setq之搭配使用]

```
(setq fl (open "C:\\sample.dat" r))
```

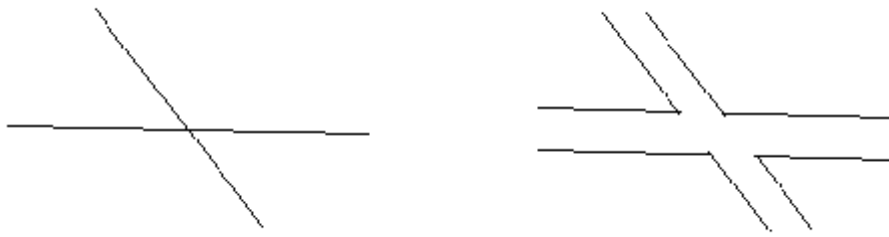
```
(setq a (read-char fl))
```

```
(close fl)
```

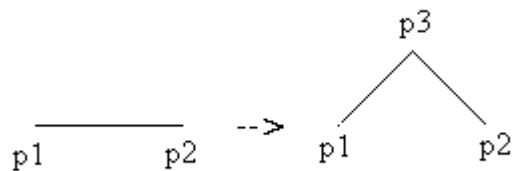
- 可以應用 AutoLisp 來產生碎形(fractal；大陸譯為分形)。

作業：【請將作業傳至 [ftlin@ccms.ntu.edu.tw](mailto:ftlin@ccms.ntu.edu.tw)】

1. 何謂 list? atom? element?
2. 一般數學函式  $g(a, h(x, y, z))$  之 Lisp 表示法為何?
3. 利用 AutoLisp 撰寫直線產生程式。  
Input：以 getpoint 指令，讓 user 輸入二個點，分別稱之為 pt1 及 pt2  
Output：繪出由 pt1 及 pt2 所構成的直線
4. 利用 AutoLisp 撰寫道路路口產生程式。  
Input：兩條交叉之道路中心線，以及其(不同道路)寬度。  
Output：道路境界線  
請列出原始程式及執行成果  
(提示：使用 mline 及 mledit 二個 AutoCAD 指令，並適當設定其引數)



5. 以 AutoLisp 產生碎形圖形，如 C-shape 或 tree。  
提示：(詳見「碎形」參考資料)



```
(defun C:Cshape(p1,p2)
  當 p1,p2 距離大於某一門檻值時
  計算 p3 坐標
  (Cshape p1 p3)
  (Cshape p3 p2)
)
```

6. 試算其結果  
A. (list '(+ 5 3) (- 8 4))  
B. (cons '(a b) '(c d e))
7. 以 setq 等指令，先建一資料表：

Name	Price	Size
BrandA	100	10
BrandA	110	12
BrandB	180	10
BrandB	200	12

以 assoc 等指令查詢 BrandB，Size=10 的 price。





;

## 名稱：亂數產生器

作者：林峰田

輸入：

1. 一個奇數，做為 seed
2. 擬生之亂數個數

輸出：一串介於 0 與 1 之間的亂數

原理：乘法同餘法亂數產生器(multiplicative congruential generator)

$$1. \quad X[n+1] = C * X[n] \bmod M$$

- (1)  $X[0]$ , 即是 seed, 必須是奇數. 本程式由使用者輸入.
- (2)  $C = 8^K + 5$ ,  $K = 0, 1, 2, 3, \dots$
- (3)  $M = 2^b$ ,  $b > 4$
- (4)  $C$  要大於  $M$
- (5) 此亂數產生器的週期為  $2^{(b-2)}$ , 亦即之後之數目會重覆循環出現
- (6) 此函式為一遞迴函式, 由  $X[i]$  產生  $X[i+1]$ ,  $i = 0, 1, 2, 3, \dots$

$$2. \quad \text{亂數 number} = X[i] / M$$

;

```
(defun C:rand()
  (setq seed (getint "input an odd number as seed --> "))
  (if (= 0 (rem seed 2))
    (princ "An odd number is needed !!!")
    (progn
      (setq times (getint "\n How many random numbers do you need? --> "))
      (setq C (expt 5 13))
      (setq M (expt 2 31))
      (repeat times
        (setq seed (rem (* C seed) M))
        (princ (setq number (abs (/ (float seed) (float M))))))
      (princ "\n")
    )
  )
)
```