个人计算机加密程序

松井甲子雄
北村孝一 合著

空军《密码信息与研究》编辑部

个人计算机加密程序

松井甲子雄
北村孝一 合著

张杭萍
徐世界
赵传友
张风民

范寅龙

一九九五年八月

前　言

随着个人计算机功能的提高及软件的日益丰富，计算机技术在实际工作中得到了广泛地运用。但你的程序及数据未必安全可靠，任何人都可以通过个人计算机查看文件，获取拷贝。随着个人信息化时代的到来，人们对计算机的安全性越来越关心。要想涉足这一领域，"数据加密"将是你必不可少的一个课题。

对计算机文件的加密感兴趣的人来说，本书是一本实习书；同时又是一本参考书，为在实际业务中对文件的安全性感到头疼的人提供了具体的方法。本书把古今中外的加密方法运用 *BASIC* 语言归结为 *94* 个程序。

本书把重点放在浅显易懂的解说程序上，如果读者还想更详细地了解加密的方法，请参考本书的姐妹篇——松井甲子雄所著《计算机加密法入门》（森北出版社出版）。总之，若本书能满足读者的求知欲望，对文件的安全管理能助一臂之力的话，我们将不胜荣幸。

作者：松井甲子雄　北村孝一

目　录

第 1 章　基本演算程序

本章介绍编写密码程序时所必需的基本技巧．首先把程序输入你的个人计算机中，边运行边阅读本书．为了使读者逐步提高，还配有简单例题，请务必弄清楚．其次，本章的程序在以下几章中都要作为子程序反复出现．所使用的语言，运用了能在日本电气公司制造的 NEC PC-8800 及 NEC PC-9800 系列上共同使用的 N_{88}-BASIC．

1. 乱数的编制方法

a. 方法

编制密码时，常常必须有乱数．个人计算机也备有乱数生成函数，请好好利用．首先必须确定乱数使用时的计算规则，这在数学上称作模（modulo）．模 7 运算只能取 0 到 6 的整数值，十进制数 10，相当于

$$10-7\equiv 3 \pmod 7$$

这样产生乱数的时候，必须指定模．下面介绍运用乱数生成函数 RND 的程序．

```
100 '##### Algorithm 1 (Ransu) #####
110 CLS:N=0:T=1
120 RANDOMIZE:PRINT                              乱数序列的初值设置
130 INPUT "モード   :  ";M:PRINT                 初值输入
140 INPUT "コスウ   :  ";P:PRINT
150 PRINT TAB(25) "<<<<<  RANSU  >>>>>"
160 PRINT
170 X=RND(1):N=N+1:R=INT(X*M)                    乱数生成(0～M-1)
180 IF T = 10 THEN T=1:PRINT
190 PRINT TAB(T*8-5) R;:T=T+1                    乱数打印
200 IF N < P GOTO 170 ELSE PRINT:PRINT
210 END
```

b. 说明

这个程序指定了 RANDOMIZE 所用的初值、模 M 及乱数的输出个数 P，这个结果产生的乱数为 0 到 M-1 的值．比如这个程序的初值是-1，M = 10，P = 10，输入后就得到下面的输出．但是本书是利用 NEC PC-9801E 来计算的，在别的计算机上会得到不同的结果．

```
Random number Seed (-32768 to 32767)? -1
模数:? 10
```

```
コスウ:? 10
                <<<<<  RANSU  >>>>>
 9  6  4  4  14  0  17
 6
OK
```

c. 注意

△关于编入函数 RND 的使用方法，请仔细阅读所用个人计算机手册.

△乱数的周期因个人计算机的机种不同而不同，使用时务必事先查明．例如在 N_{88}-BASIC (PC-9801) 上，当 RANDOMIZE-1 时，得到的周期为 262144.

2. 乱数序列的生成

a. 方法

把字母序列（或数字）打乱顺序所得到的排列称乱数序列．乱数序列在加密时往往是必不可少的．对 26 个字母来说，生成乱数序列的方法是，生成乱数后作 mod 26 运算，转换成文字并排列．但是当相同的字母已得到记录时就要舍去，而生成一个乱数．其程序如下所示．

```
100 '##### Algorithm 2 (Ranji) #####
110 CLS:RANDOMIZE:PRINT                          乱数序列的初值设置
120 INPUT "How many groups  ";N:PRINT            组数输入
130 DIM KT(25,N)
140 FOR I=1 TO N
150 L=0:X=RND(1):Y=INT(X*26)                     第 1 组乱数(模 26)
160 KT(0,I)=Y:L=L+1
170 WHILE L < 26
180 X=RND(1):Y=INT(X*26)
190 FOR J=0 TO L-1                               第 2～第 26 组乱数的比较
200 IF KT(J,I)=Y GOTO 230
210 NEXT J
220 KT(L,I)=Y:L=L+1
230 WEND
240 NEXT I:PRINT
250 FOR J=1 TO N
260 PRINT "[";J;"]";TAB(9)
270 FOR I=0 TO 25
280 PRINT CHR$(KT(I,J)+65);" ";                  输出(大写英文)
290 NEXT I:PRINT:PRINT
300 NEXT J:PRINT
310 END
```

b. 说明

这个程序运用了英文 26 个字母，由于第二组以后的字母是否有重复都要进行校对，所以存在费时的缺点。此程序一运行，即可在排列 KT 中随机产生 N 组英文字母列，例如，当 N＝2 时，如下所示。

```
Random Number Seed (-32768 to 32767)? 25
How Many groups ? 2
(1)  YRGMEZVFQLNIASPDUWKOCTBJHX
(2)  CIYFHETZPLQDRVMXOSWBKGUJNA
OK
```

c. 注意

在 280 行的 PRINT 语句中有：

```
PRINT CHR$ (KT(I,J)+65)
```

这是为了输出大写字母而加上了 $65_{(10)}(=41_{(16)})$。输出小写字母时，就要变成：

```
PRINT CHR$ (KT(I,J)+97)
```

3. 文字的二进制表示

a. 方法

从个人计算机的键盘输入数据时，文字、数字和特殊符号用 ASCII 码（表 1.1）来表示。因而，每个比特加密的情况下，ASCII 码（十进制）必须用 8 位二进制表示，这只要通过除 2 求余的基本操作就行了。这种方法是字母加密经常使用的技巧之一，其程序如下所示。

```
100 '##### Algorithm 3 (Binarization) #####
110 DIM X(9),Y(8):CLS
120 PRINT "Please input any key ··· "          输入(英文数字)
130 Ps=INKEY$:IF Ps="" GOTO 130
140 X(1)=ASC(Ps)
150 FOR I=1 TO 8
160 X(I+1)=INT(X(I)/2)                         二进制操作
170 Y(I)=X(I)-X(I+1)*2
180 NEXT I
190 PRINT:PRINT "[";Ps;"]";                    输出(二进制数)
200 PRINT "  &H";HEX$(X(1));" : " ;
210 FOR I=8 TO 1 STEP -1:PRINT Y(I);:NEXT I
220 PRINT:PRINT
230 GOTO 120
240 END
```

表 1.1 ASCII 码

0		44	,	88	X	132	▬	176	−	220	ワ				
1	SH	45	-	89	Y	133	▬	177	ア	221	ン				
2	SX	46	.	90	Z	134	▬	178	イ	222	゛				
3	EX	47	/	91	[	135	▬	179	ウ	223	°				
4	ET	48	0	92	¥	136	▏	180	エ	224	=				
5	EQ	49	1	93	]	137	▏	181	オ	225	ト				
6	AK	50	2	94	^	138	▏	182	カ	226	キ				
7	BL	51	3	95	_	139	▏	183	キ	227	ヰ				
8	BS	52	4	96	`	140	▏	184	ク	228	◢				
9	HT	53	5	97	a	141	▏	185	ケ	229	◣				
10	LF	54	6	98	b	142	▏	186	コ	230	◥				
11	HM	55	7	99	c	143	┼	187	サ	231	◤				
12	CL	56	8	100	d	144	┴	188	シ	232	♠				
13	CR	57	9	101	e	145	┬	189	ス	233	♥				
14	SO	58	:	102	f	146	┤	190	セ	234	◆				
15	SI	59	;	103	g	147	├	191	ソ	235	♣				
16	DE	60	<	104	h	148	▔	192	タ	236	●				
17	D1	61	=	105	i	149	─	193	チ	237	○				
18	D2	62	>	106	j	150	│	194	ツ	238	／				
19	D3	63	?	107	k	151	┘	195	テ	239	＼				
20	D4	64	@	108	l	152	┌	196	ト	240	×				
21	NK	65	A	109	m	153	┐	197	ナ	241	円				
22	SN	66	B	110	n	154	└	198	ニ	242	年				
23	EB	67	C	111	o	155	┘	199	ヌ	243	月				
24	CN	68	D	112	p	156	╱	200	ネ	244	日				
25	EM	69	E	113	q	157	╲	201	ノ	245	時				
26	SB	70	F	114	r	158	╲	202	ハ	246	分				
27	EC	71	G	115	s	159	ノ	203	ヒ	247	秒				
28	→	72	H	116	t	160		204	フ	248					
29	←	73	I	117	u	161	・	205	ヘ	249					
30	↑	74	J	118	v	162	「	206	ホ	250					
31	↓	75	K	119	w	163	」	207	マ	251					
32		76	L	120	x	164	、	208	ミ	252					
33	!	77	M	121	y	165	・	209	ム	253					
34	"	78	N	122	z	166	ヲ	210	メ	254					
35	#	79	O	123	{	167	ァ	211	モ	255					
36	$	80	P	124	¦	168	ィ	212	ヤ						
37	%	81	Q	125	}	169	ゥ	213	ュ						
38	&	82	R	126	~	170	ェ	214	ヨ						
39	'	83	S	127		171	ォ	215	ラ						
40	(	84	T	128	_	172	ャ	216	リ						
41	)	85	U	129	▬	173	ュ	217	ル						
42	*	86	V	130	▬	174	ョ	218	レ						
43	+	87	W	131	▬	175	ッ	219	ロ						

b. 说明

在 150～180 行中输入的字母转换为二进制．比如输入"A"和"P"时，输出就会显示如下内容．

```
Plaese input any key...
(A)   &H41: 0  1  0  0  0  0  0  1
Plaese input any key...
(P)   &HB1: 1  0  1  1  0  0  0  1
```

c. 注意

ASCII 码中，从 0 到 $31_{(10)}$的值是作为控制符来使用的，在屏幕上显示时要多加注意．

4. 数字的二进制表示

a. 方法

为了用比特单位处理十进制的数值数据，必须变换成二进制数．其变换方法虽然通过除 2 求余数就足够了，但下面的程序也能同时求出二进制的位数．

```
100 '##### Algorithm 4 (Decimal to Binary) #####
110 CLS
120 INPUT"Input decimal number [MAX:32767] ";DEC    输入(十进制数)
130 PRINT:A=DEC:CT=1:C$=""
140 WHILE A > 1
150 CT=CT+1:B=INT(A/2):C=A MOD 2                    十进制～二进制变换
160 C$=RIGHT$(STR$(C),1)+C$:A=B
170 WEND
180 C$=RIGHT$(STR$(A),1)+C$
190 PRINT "# decimal = ";TAB(12) DEC;TAB(30) "#"
200 PRINT "# binary  = ";TAB(13) C$;TAB(30) "#"     输出(二进制)
210 PRINT "# bit     = ";TAB(12) CT;TAB(30) "#"
220 PRINT:GOTO 120
230 END
```

b. 说明

当该程序输入 0～32767 范围内的十进制数时，能将其值换成二进制数并显示其位数．比如输入 127 时得到如下输出．

```
Input decimal number (MAX:32767)? 127
# decimal  =  127                  #
# binary   =  1111111              #
# bit      =  7                    #
```

c. 注意

在密码领域里不使用负数和实数，这是众所周知的常识。如遇到负值就请在运算上加上模变为正值。比如：

$$-7\equiv -7+26\equiv 19 \pmod{26}$$

mod 26 中规定 A＝0，B＝1，……，Z＝25，于是 19 则表示字母 T。

5. 密钥的整序

a. 方法

加密时要用单词、成语或短句等文字列作为密钥，密钥所含的文字称为密钥字。数据加密时，要把密钥字按字母顺序从大到小或从小到大重新排列，其方法是多种多样的。下面的程序运用了最基本的大小替换方式。

```
100 '##### Algorithm 5 (Ksort) #####
110 CLS:PRINT "Input characters  [max:255]"          字符系列输入
120 PRINT:LINE INPUT "> ";A$:PRINT
130 N=LEN(A$)
140 DIM [DATA(N),L(N)
150 FOR I=1 TO N
160 [DATA(I)=ASC(MID$(A$,I,1)):L(I)=I                码变换
170 NEXT I
180 L1=N-1
190 FOR I=1 TO L1
200 K=I+1
210 FOR J=K TO N
220 IF [DATA(I) > [DATA(J) OR L(I) > L(J) THEN
    M1=[DATA(I):M2=L(I):[DATA(I)=[DATA(J):           检索操作
    L(I)=L(J):[DATA(J)=M1:L(J)=M2
230 NEXT J
240 NEXT I
250 PRINT "<<<<<  Input  character  >>>>> "          输出 1
260 PRINT:T=0
270 FOR I=1 TO N STEP 5
280 IF T=10 THEN T=0
290 PRINT TAB(T*6) MID$(A$,I,5);" ";:T=T+1
300 NEXT I:PRINT:PRINT
310 PRINT "<<<<<  Sorted  character  >>>>>"          输出 2
320 PRINT:T=0
330 FOR I=1 TO N
340 IF T=15 THEN T=0
350 PRINT TAB(T*4) CHR$([DATA(I));:T=T+1
360 NEXT I:PRINT:PRINT
370 PRINT "<<<<<  Character position before ";       输出 3
380 PRINT "being sorted  >>>>>":PRINT:T=0
390 FOR I=1 TO N
400 IF T=15 THEN T=0
410 PRINT TAB(T*4) L(I);:T=T+1
420 NEXT I:PRINT:PRINT
430 END
```

b. 说明

该程序最多可输入 255 个英文字母．190～240 行进行整序操作．该程序运行后，可按 ABC……的顺序输出如下重新排列的密钥，而且清楚地显示出各个密钥字输入时的先后顺序．

```
Input characters [max: 255]
> COMPUTER
<<<<< Input character >>>>>
COMPU TER
<<<<< Sorted character >>>>>
C E M O P R T U
<<<<< Character position before being sortde >>>>>
1 7 3 2 4 8 6 5
OK
```

c. 注意

该程序采取对于相同字母，先输入的优先编号的方法．但是根据不同情况，有时把密钥文序列中第二次以后出现的字母都删去再整序．在下面各章中使用时要注意这一点．

6. 密钥的搜索

a. 方法

假定密钥字序列为一次元排列，当欲知要寻找的字母在排列中所处位置时，就可使用此程序．此程序采取从开始按顺序比较的方法，这种方法最简单．这时一般假定密钥的长度很短，若密钥较长时，就在程序中采用效率高的二分搜索法．

```
100 '##### Algorithm 6 Ksearch #####
110 DIM KT(26)
120 GOSUB *TABLE:CLS
130 PRINT "Input character to be searched : "; 英文输入
140 LINE INPUT C$:PRINT:PK=ASC(C$)-64
150 GOSUB *KSEARCH
160 PRINT "position of [ ";C$;" ] : ( No.";
170 PRINT IP;")":PRINT:PRINT
180 GOTO 130
190 '
200 *KSEARCH
210 FOR I=1 TO 26
220 IF PK=KT(I) THEN IP=I:GOTO 250                    表内对照搜索
230 I=0
```

```
240 NEXT II
250 RETURN
260 '
270 *TABLE
280 FOR I=1 TO 26
290 READ KTC$                                    表输入
300 KT(I)=ASC(KTC$)-64
310 NEXT I
320 '========  Table  data  ========
330 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,              密钥序列
        N,O,P,Q,R,S,T,U,V,W,X,Y,Z
340 '================================
350 RETURN
```

b. 说明

用作比较的密钥要预先在排列 KT 中下定义．在该程序中，26 个英文字母在 DATA 语句（330 行）内给定．于是，应该查找的密钥字一输入 PK（140 行）时，就可输出它在表格内的位置．下面是一例子．

```
Input character to be searched : C
position of  [ C ] : ( No. 3 )
Input character to be searched : P
position of  [ P ] : ( No. 16 )
```

c. 注意

一旦象该程序这样将密钥字列设定为比较表，到最后密钥情报就会残留在程序内．所以在使用中这一点请注意．当然，密钥字序列每次从键盘输入比较安全．

个人计算机安全保护子菜单（1）

计算机系统正面临着各种病毒的威胁．下面就介绍几种暗中活动的病毒手法．

(1) 特洛伊木马

这种手法用在公司重要程序里．非法侵入者插入特殊命令后，在公司业务正常运行的同时，通过改写也能达到非法入侵者的目的．

(2) 陷井入口

这是一种以调试程序为目的而设置的陷井入口程序．非法入侵者进行存取，在程序内搜索，捕获出密码密钥、UID 番号及口令．

第 2 章　代替式密码程序

本章介绍代替作业密码的编制，这是最基本的加密法．各种作业方法的程序分为加密程序和脱密程序．两者存在互为逆变换关系，因此这里主要说明加密．另外，加密程序在其程序名后标上 E(encryption)，脱密程序在其程序名后标上 D(decryption)，以示区别．而且这里叙述的程序，为了便于理解起见，进行加密的数据采用了从键盘输入的方法．但实际运用中，程序及数据都贮存在文件内，因此输入数据时，要参照汇总在卷末附录中数据读入的例子加以修改．

7. 凯撒密码

a. 加密

所谓凯撒密码，即为在输入字母 P_i 上加某一个数（密钥）K 后，输出字母 C_i 的单码代替方式．这种密码按下面的式子简单生成．

$$C_i \equiv P_i + K \pmod{M}$$

这里输入的数据为一般文章，故称之为明码（或明文），而输出的数据为经过加密而变成看不懂的字母序列，故称之为密码（或密文）．下面就是加密程序．

```
100 '##### Algorithm 7E (Caeser) #####
110 CLS:PRINT "<<<  Caeser encryption  >>>"
120 DIM PDATA(255),CDATA(255)
130 PRINT:LINE INPUT "File name -> ";F$:PRINT          密文输出用文件名
140 OPEN F$ AS #1:RN=0
150 FIELD #1,1 AS N$,255 AS BUF$
160 INPUT "Input shift value ";K                       密钥输入
170 CLS:PRINT "Input plaintext [max:255]":PRINT
180 LINE INPUT "> ";P$:N=LEN(P$):PRINT:PRINT           明文输入
190 FOR I=1 TO N
200 PDATA(I)=ASC(MID$(P$,I,1))                         码变换
210 NEXT I
220 FOR I=1 TO N
230 CDATA(I)=(PDATA(I)+K) MOD 256                      加密运算
240 NEXT I
250 GOSUB *STORE1
260 PRINT"### Continue plaintext  [Y or N] ### ?"     是否继续输入
270 Y$=INKEY$:IF Y$="" GOTO 270
280 IF Y$="Y" OR Y$="y" THEN 170
290 IF Y$="N" OR Y$="n" GOTO 300 ELSE 270
300 CLOSE #1:PRINT:PRINT "< END >":PRINT
310 END
320 '
330 *STORE1
340 CD$=""                                             加密数据的输出
350 FOR I=1 TO N
360 C$=CHR$(CDATA(I)):CD$=CD$+C$
```

```
370 NEXT I
380 LSET BUF$=CD$:LSET N$=CHR$(LEN(CD$))
390 RN=RN+1:PUT #1,RN
400 RETURN
```

b. 说明

该程序假定输入的明码由 ASCII 码组成（200 行），故而，从键盘输入任意一个字母，程序就会按密钥 K 指定的差数变成另一字母输出。

此程序输入的明码（180 行）规定在 255 个字母内分段加密。如果数据量超过 255 个字就依照 260 行的指示，输入 Y（继续）或 N（结束）。这样加密后的数据就由子程序 STORE 1 写入文件。此文件名必须先在 130 行输入。下面为输入输出的例子。

```
<<<  Caeser encryption  >>>
File name -> TX7
Input shift value ? 144
Input plaintext [max:255]
> ｱﾅﾀ ﾊ ｼｰｻﾞｰｱﾝｺﾞｳ ﾖ ﾘｶｲ ﾃﾞｷﾏｼﾀｶ
<<<<<   PLAINTEXT   >>>>>
ｱﾅﾀ ﾊ ｼｰｻﾞｰｱﾝｺﾞｳ ﾖ ﾘｶｲ ﾃﾞｷﾏｼﾀｶ
<<<<<   CIPHERTEXT   >>>>>
AUP-Z-L@Kn@AmJnC-6-hFB-SnG_LPF
### Continue plaintext  [Y or N] ### ? N
< END >
Ok
```

c. 注意

△输入的数据可以作为英文(mod 26)及片假名(mod 64)加以处理，这时要把 mod 256 运算换成各自的模。比如英文字母按下式变化。

```
200  PDATA(I) = ASC(MID$(P$ , I , 1)) - 65
230  CDATA(I) = (PDATA(I) + K)  MOD  26
360  C$ = CHR$(CDATA(I) + 65)
```

△要加密的程序或数据如果已在文件中，请按附录所示方法写入所输数据。

△正如上例所示，密码一般比较难读懂，为简单明了起见，习惯上采用每5个字为一段的方法输出。本书下面也尽量沿用这一习惯。

d. 应用

将该程序中的 230 行也可变成如下的形式:

230 CDATA(I)=(PDATA(I)＊K) MOD M

但是，这时选择的 K 值与 M 要互质（无公约数），否则，就会出现周期性，这一点请多加注意。

e. 脱密

下列程序将由凯撒密生产的密文翻译成明文。从 130 行指定的文件中读入密码，然后对每个字母进行逆变换。此时，当 240 行的 PDATA(I) 变成负值时，250 行就换成正值。明文在屏幕上输出，可用子程序＊DISPLAY256 D 来显示。用 mod 256 时，若 PDATA(I)比 $31_{(10)}$小的话，它就变成控制码，屏幕显示会出现异常，所以要多加小心。若用 mod 26，子程序就要改用＊DISPLAY26D。

```
100 '##### Algorithm 7D (Caeser) #####
110 CLS:PRINT "<<<  Caeser decryption  >>>"
120 DIM PDATA(255),CDATA(255)
130 PRINT:LINE INPUT "File name -> ";F$:PRINT          密文输入用文件名
140 OPEN F$ AS #1:RN=LOF(1)
150 FIELD #1,1 AS N$,255 AS BUF$
160 INPUT "Input shift value ";K:PRINT                 密钥输入
170 CLS:PRINT "<<<<<   PLAINTEXT   >>>>>"
180 FOR REC=1 TO RN
190 GET #1,REC:CD$=LEFT$(BUF$,ASC(N$)):N=LEN(CD$)      从磁盘读入密文
200 FOR I=1 TO N
210 CDATA(I)=ASC(MID$(CD$,I,1))                        码变换
220 NEXT I
230 FOR I=1 TO N
240 PDATA(I)=(CDATA(I)-K) MOD 256                      脱密运算
250 IF PDATA(I) < 0 THEN PDATA(I)=PDATA(I)+256
260 NEXT I
270 GOSUB ＊DISPLAY256D
280 NEXT REC
290 CLOSE #1:PRINT:PRINT "< END >":PRINT
300 END
310 '
320 ＊DISPLAY256D
330 PD$=""                                             明文打印
340 FOR I=1 TO N
350 C$=CHR$(PDATA(I)):PD$=PD$+C$
360 NEXT I
370 PRINT:T=0
380 FOR I=1 TO N
390 IF T=60 THEN T=0:PRINT
400 T$=MID$(PD$,I,1):PRINT T$;:T=T+1
410 NEXT I:PRINT
420 RETURN
```

8. 密钥错乱式单码代替密码

a. 加密

这种密码依照密钥字列中密钥字的字母顺序，替换 26 个英文字母进行排列（称这种操作为按密钥字进行的转置）。比如密钥是"COMPUTER"（1 2 3 4 5 6 7 8），在 410 行到 460 行内，从排列 L (I) 中求得密钥字的先后顺序为"17324865"。接着，490 行到 540 行把 26 个英文字母按 8 个字母一行，排在 DUMMY (I, J) 中，按 L (I) 的顺序从 J 列取出，并设定代替表 KC (L1)，其结果如下：

明：ABCDEFGHIJKLMNOPQRSTUVWXYZ
密：AIQYGONCKSBJRZDLTHPXFNVEMU

该程序用这张代替表，把输入的每个文字（明码）换成输出文字（密码）。

```
100 '##### Algorithm 8E (KTrans) #####
110 CLS:PRINT "<<<  KTrans encryption  >>>"
120 DIM PDATA(255),CDATA(255)
130 PRINT:LINE INPUT "File name -> ";F$:PRINT        密文输出用文件名
140 OPEN F$ AS #1:RN=0
150 FIELD #1,1 AS N$,255 AS BUF$
160 GOSUB *KCTABLE
170 CLS:PRINT "Input plaintext  [max:255]"           明文输入
180 PRINT:LINE INPUT "> ";P$:N=LEN(P$)
190 FOR I=1 TO N
200 PDATA(I)=ASC(MID$(P$,I,1))-65                    码变换
210 NEXT I
220 FOR I=1 TO N
230 CDATA(I)=KC(PDATA(I))                            单码代替处理
240 NEXT I
250 GOSUB *STORE2
260 GOSUB *DISPLAY26E
270 PRINT"### Continue plaintext [Y or N] ### ?"    是否继续输入
280 Y$=INKEY$:IF Y$="" GOTO 280
290 IF Y$="Y" OR Y$="y" GOTO 170
300 IF Y$="N" OR Y$="n" GOTO 310 ELSE GOTO 280
310 CLOSE #1:PRINT:PRINT "< END >":PRINT
320 END
330 '
340 *KCTABLE
350 LINE INPUT "Input keyword : ";K$:N=LEN(K$)       密钥输入
360 DIM IDATA(N),L(N)
370 FOR I=1 TO N
380 IDATA(I)=ASC(MID$(K$,I,1))-65:L(I)=I             码变换
390 NEXT I
400 L1=N-1
410 FOR I=1 TO L1
420 K=I+1
430 FOR J=K TO N
```