50116 Program Description I

Program Title CODER-DECODER

Contributor's Name Andrew M Stephenson

Address 19 Du Pre Walk, Wooburn Green, High Wycombe, Bucks.

City Country U.K. Postal Code HP10 0QJ

Program Description, Equations, Variables Inspired by the "Enigma" coding machine used by the German Government before and during WW2, but actually more similar to their "Geheimschreiber", the program encodes and decodes integer numbers lying in the range 00-99 under the control of five pseudorandom number generators based on keys stored in registers R_A to R_E. Cycle length, if based on HP p.n.g. which starts its sequence with ·5284163, is 500,000 for each key; and each may be drawn from different points along the same overall sequence---routines fd and fe will "exercise" the sequence continuously, thereby advancing it as far as desired.

ALGORITHMS: *Encoded = (Uncoded + Datashift) --- with the 100s removed.*

Uncoded = (Encoded - Datashift + 100) --- with the 100s removed.

To compute datashift value:

1: Compute first Keyshift value; = INT(5xFRAC(997xKey"A")). The value (0 to 4) directs the program to the next Key for the next Keyshift operation: 0-4::A-E.

2: Compute the remainder of the Keyshifts programmed for. There is capacity for up to seven in all, although only six have been shown in the prgm listing. Note that no Keyshifts need be programmed for; prgm will then use only Key"A" when generating Datashift values. Keys advance after use.

3: Last Keyshift dictates which Key is used for Datashift (Key"n"):

Datashift = INT(100xFRAC(997xKey"n")).

Operating Limits and Warnings No code is perfect; and some are less perfect than others. Before entrusting important information or numerical data to this encoding system, the user should take expert advice. To improve security, these precautions should be observed: 1)Keep messages as brief and to the point as possible; 2)Avoid use of "obvious" or "probable" message phrases, as well as standard practice; 3)Never keep keys (whether on cards or on paper) with encoded messages; 4)Use more than one code group for common letters, symbols, etc.; 5)Don't get careless, as we all do.

This program has been verified only with respect to the numerical example given in *Program Description II.* User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP NOR THE CONTRIBUTOR MAKES ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND WITH REGARD TO THIS PROGRAM MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER HP NOR THE CONTRIBUTOR SHALL BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, USE OR PERFORMANCE OF THIS PROGRAM MATERIAL.

TYPICAL APPLICATION:

Preamble: *Record on a data card: all registers zero except*

R_A=.5284163 R_B=.6298919 R_C=.5177719 R_D=.5684759 R_E=.2079123
This card is the "Key Card" and is used in the main procedure, below.

Main Procedure:

Ace industrial spy Boris Espion wishes to transmit to his superiors in Vulgaria the identity of a new coding machine, the "HEWLETT-PACKARD HP-67". The letters he transliterates into numbers, using the relationship A-Z (conventional sequence):: 01-26; the numbers he writes as 0-9::30-39, and "-" and "space" as 40 and 41 respectively. To be specially cunning, he writes the duplicate letters with '50' added. Thus his outgoing message in plain language reads:

08 05 23 12 55 20 70 40 16 01 03 11 51 18 04 41 58 66 90 36 37 00

Loading the program card, he then loads the Key Card for the day and proceeds to press "E", thereby selecting "Encode" mode. Then he presses "B" and enters each of his message numbers in turn at each PAUSE; after the last one he waits for the calculator to pause then presses: "GTO f c" and loads a blank card into the card reader. This he despatches to his superiors.

To ensure that he has a copy of the encoded message, he also presses "fB": this causes his HP-67 to display registers R_0 through to R_{19}, but since he has only required R_0 to R_4, he presses "R/S" after the fifth display. One day, he promises himself, he'll apply for an HP-97, which will save him the bother of copying down the values. This is what he gets:

0.582329548
9.763466875
9.454124987
6.054664056
9.150

A couple of facts are worth noting about these figures: first, they are held in the calculator stores as values one-tenth of these (eg: .0582329548); second, Boris is a very lax operator--he should have added some meaningless groups after the "00" to confuse the enemy.

To be extra sure he has it right, Boris reloads the Key Card, presses "D" (for Decode), followed by "C", loads message card during PAUSE, then waits for the calculator to decode the message which he then displays using "fB". Note that it halted after decoding the "00" group. Sure enough, he gets the original groups.

Later that same day Boris receives his reply:

06 81 68 98 13 77 68 02 91 91

As this is such a short message, Boris decodes it by loading the Key Card, pressing "D", then writing each group in turn followed by "A" (after each): each group is then decoded and presented in the display separately.

Boris, poor man, cannot understand his superiors' meaning.

Practical Note: *The conversion time is critically dependent on the number of keyshifts incorporated into Routine A. When using Routine C, the following approximate times apply to a full conversion of all 100 groups that can be held in store at one time:*

2 keyshifts	*:*	*15 minutes*
4 "	*:*	*20 "*
6 "	*:*	*25 "*

Keys: *Ideally, a sequence generated by a key should be as close to random as possible. In practice, the desired quality is 'obscurity', or a variability that the unauthorised recipient of a message cannot be expected to predict without access to the key or keys used to encode the message. Life may be made that much harder for such people by multiple encoding, and general rearrangement of the encoded groups. Whatever course is adopted by the user should, naturally, be kept secret.*

USERS' CLUB HP EUROPE

50116 User Instructions

CODER-DECODER				
CONVERT x	CONV.x & ACCUMULATE	CONVERT R_0--R_{19}	DECODE	ENCODE
REVIEW A-E	RVW R_0-R_{19}	RCRD DATA	ADVANCE KEY IN x REG.	

1 ◀ ▶ 2

f {

STEP	INSTRUCTIONS	INPUT DATA/UNITS	CONTROLS		OUTPUT DATA/UNITS
	NOTE: Throughout the text of this documentation,				
	"keys" is used solely in the cyphering sense.				
1	Load both sides of program card.				
2a	To select "Decode" mode:—	any	D		no change
2b	To select "Encode" mode:—	any	E		no change
3	Load initial Keys A-E either manually or from				
	a prerecorded data card. Keys B-E (any or all)				
	can be "0" but if any are zero, Routine "A" must				
	embody an even number of "Keyshift" operations.				
4a	To convert values piecemeal:—	value	A		conv.value
4b	To convert values, entered one-by-one, and				
	accumulate them sequentially in regs. R_0-R_{19}:—	none	B		PAUSE:"0"
	During PAUSE, write the first value—	value	wait		etc.
	Prgm pauses repeatedly; if no value is				
	entered, prgm assumes "0". When all of regs				
	R_0-R_{19} are full (or, actually, overwritten),				
	prgm carries out Step 7, below.				
4c	To convert values held in R_0-R_{19}:—	none	C		PAUSE:"19"
	Prgm pauses (repeatedly) until a data card				
	(which will define only R_0-R_{19}) is loaded or				
	any number key is pressed. It then works				
	through all of R_0-R_{19}, number pair by number				
	pair, then carries out Step 7, below. Prgm				
	halts after encoding "00" or after decoding				
	to yield "00". If more values are to be				
	converted, press "R/S"; prgm will halt again				
	when the next "00" is encountered. To go at				
	once to step 7, below, press: "GTO f c".				
5	To review (display/print) Keys A-E:—	none	f	A	
6	To review (display/print) R_0-R_{19} (multiplied by				
	10 for formatting reasons):—	none	f	B	
7	To record contents of R_0-R_{19} on a data card,				
	having zeroed R_A-R_E&R_I (but saving Keys A-E):—	none	f	C	"Crd"
	If this routine is called as a result of				
	steps 4b or 4c and no card is to be recorded:--		R/S	R/S	
8	To run a Key value ahead through its sequence:—	Key	fD or fE		{ see text
	To halt (prgm will run indefinitely):—		R/S		!!any!!
	Register I is incremented for each full cycle				
	completed. Single-step to end of a cycle.				

50116 Program Listing I

Page 4 of 5

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	f LBL A	31 25 11	
	DSP 0	23 00	
	2	02	
	0	00	
	h x⇄I	35 24	
	h x⇄y	35 52	
	f GSB 0	31 22 00	=keyshift 1 Quantity of keyshifts should be determined by the user's needs.
	f GSB 0	31 22 00	=keyshift 2
	f GSB 0	31 22 00	=keyshift 3
010	f GSB 0	31 22 00	=keyshift 4
	f GSB 0	31 22 00	=keyshift 5
	f GSB 0	31 22 00	=keyshift 6
	h SF 0	35 51 00	
	f GSB 0	31 22 00	=datashift
	h CF 0	35 61 00	
	h RCI	35 34	
	h F? 1	35 71 01	="decoding?"
	CHS	42	
	+	61	
020	E EX	43	
	2	02	
	h STI	35 33	
	+	61	
	h RCI	35 34	
	÷	81	
	g FRAC	32 83	
	h RCI	35 34	
	×	71	
	h x y	35 52	
030	h STI	35 33	
	h R↓	35 53	
	h RTN	35 22	
	f LBL B	31 25 12	
	0	00	
	GTO 2	22 02	
	f LBL 1	31 25 01	
	h RCI	35 34	
	.	83	
	2	02	
040	+	61	
	f LBL 2	31 25 02	
	h STI	35 33	
	2	02	
	0	00	
	g x=y	32 51	
	GTO f c	22 31 13	
	h CF 3	35 61 03	
	0	00	
	h PAUSE	35 72	(enter data value)
050	h F? 3	35 71 03	
	+	61	
	f GSB A	31 22 11	
	h RCI	35 34	
	g FRAC	32 83	
	1	01	
	0	00	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	×	71	
	f x=0	31 51	
	STO (i)	33 24	
060	2	02	
	+	61	
	g 10^x	32 53	
	÷	81	
	STO + (i)	33 61 24	
	GTO 1	22 01	
	f LBL C	31 25 13	
	h CF 3	35 61 03	
	1	01	
	9	09	
070	h STI	35 33	
	g MERGE	32 41	
	h PAUSE	35 72	(load data card)
	0	00	
	h F? 3	35 71 03	
	GTO 4	22 04	
	GTO C	22 13	
	f LBL 3	31 25 03	
	h RCI	35 34	
	.	83	
080	2	02	
	+	61	
	f LBL 4	31 25 04	
	h STI	35 33	
	2	02	
	0	00	
	g x=y	32 51	
	GTO f c	22 31 13	
	h RCI	35 34	
	g FRAC	32 83	
090	1	01	
	0	00	
	×	71	
	g 10^x	32 53	
	STO × (i)	33 71 24	
	RCL (i)	34 24	
	g FRAC	32 83	
	E EX	43	
	2	02	
	STO × (i)	33 71 24	
100	×	71	
	f INT	31 83	
	STO − (i)	33 51 24	
	f x=0	31 51	
	h SF 2	35 51 02	
	h F? 1	35 71 01	="decoding?"
	h CF 2	35 61 02	
	f GSB A	31 22 11	
	STO + (i)	33 61 24	
	h RCI	35 34	
110	g FRAC	32 83	
	1	01	
	0	00	

REGISTERS

Registers R_0 through to R_{S9} are reserved for storage of data values in groups of 5.

A	B	C	D	E	I
Key A	Key B	Key C	Key D	Key E	used

EUROPE · 5 0 1 1 6

Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
	×	71			0	00	
	2	02		170	×	71	
	+	61			f -x-	31 84	
	g 10x	32 53			2	02	
	STO ÷ (i)	33 81 24			0	00	
	h R↓	35 53			f ISZ	31 34	
	h F? 2	35 71 02			h RCI	35 34	
120	R/S	84			g x≠y	32 61	
	f x≠0	31 61			GTO 5	22 05	
	GTO 3	22 03			DSP 0	23 00	
	h F? 1	35 71 01			h RTN	35 22	
	R/S	84		180	g LBLf c	32 25 13	=record data card
	GTO 3	22 03			0	00	
	f LBL 0	31 25 00	=shift routine		h STI	35 33	
	RCL (i)	34 24			RCL E	34 15	
	9	09			+	61	
	9	09			RCL A	34 11	
130	7	07			RCL B	34 12	
	×	71			RCL C	34 13	
	g FRAC	32 83			RCL D	34 14	
	STO (i)	33 24			h x⇄I	35 24	
	5	05		190	STO A	33 11	
	×	71			STO B	33 12	
	2	02			STO C	33 13	
	0	00			STO D	33 14	
	h F? 0	35 71 00	="datashift?"		STO E	33 15	
	×	71			h x⇄I	35 24	
140	h F? 0	35 71 00	="datashift?"		f W/DATA	31 41	(run card)
	0	00			h R↑	35 54	
	+	61			STO A	33 11	
	f INT	31 83			h R↑	35 54	
	h STI	35 33		200	STO B	33 12	
	h R↓	35 53			h R↑	35 54	
	h RTN	35 22			STO C	33 13	
	g LBLf a	32 25 11	=key review		h R↑	35 54	
	DSP 7	23 07			STO D	33 14	
	RCL A	34 11			h LSTx	35 82	
150	f -x-	31 84			STO E	33 15	
	RCL B	34 12			CLx	44	
	f -x-	31 84			h RTN	35 22	
	RCL C	34 13			g LBLf d	32 25 14	=advance Key in x
	f -x-	31 84		210	g LBLf e	32 25 15	along sequence.
	RCL D	31 14			9	09	
	f -x-	31 84			9	09	
	RCL E	34 15			7	07	
	f -x-	31 84			×	71	
	0	00			g FRAC	32 83	
160	DSP 0	23 00			f ISZ	31 34	
	h RTN	35 22			GTO f d	22 31 14	
	g LBLf b	32 25 12	=data review		f LBL D	31 25 14	=set "decode"
	DSP 9	23 09			h SF 1	35 51 01	
	0	00		220	h RTN	35 22	
	h STI	35 33			f LBL E	31 25 15	=set "encode"
	f LBL 5	31 25 05			h CF 1	35 61 01	
	RCL (i)	34 24			h RTN	35 22	
	1	01					

LABELS					FLAGS
A CONVERT "x"	B CONVERT "x" & ACCUMULATE.	C CONVERT $R_0 \rightarrow R_{59}$ (R_{19})	D SET "DECODE"	E SET "ENCODE"	0 DATASHIFT?
a REVIEW CODE KEYS	b REVIEW $R_0 - R_{19}$	c RECORD R_0 – R_{19} ON CARD	d ADVANCE KEY	e IN x REGISTER	1 DECODE?
0 used	1 used	2 used	3 used	4 used	2 used
5 used	6	7	8	9	3 used

SET STATUS

FLAGS	ON	OFF	TRIG		DISP	
0	□	■	DEG	□	FIX	■
1	□	■	GRAD	□	SCI	□
2	□	■	RAD	□	ENG	□
3	□	■			n 0	