* TI PPC NOTES * V7n6, 1982

NEWSLETTER OF

THE TI PROGRAMMABLE CALCULATOR CLUB

9213 Lanham Severn Road

Lanham MD 20706, USA.

Many of you have written me lately, asking me to print a simple, yet efficient index for the two years of our existence. To tell you the truth, I have had such an index all along, stored on diskettes for my TI-99/4A computer. One list was sorted according to subject matter, which I called ARTICLE and a second one according to AUTHOR. Most of the time, that is all what is needed to find a specific article written by a specified author. The computer doesn't take more than three seconds to find what I want. So, with this issue you'll find the print-out of both indices, one for 1980 and one for 1981. This way, you can file each loose leaf at the head of its corresponding volume.

The actual printing was done on an Olympia electronic typewriter (and so is this editorial) pasted up and subsequently reduced to 66% of full size, so as to fit on an 8.5 by 11 inches sheet. (to my European friends that is a DIN A4 sheet. And you thought I didn't know about those things, did you?)

Instead of the usual 16 we have 18 pages this time. This to accomodate the extended treatment of the now famous Zimmerman puzzle. This challenge turned out to be a real success: so far (May 26th) I received 31 solutions, and all of them excellent! Starting on page 15 you will find the first 14 of them.

There is lots of expectation and many rumors about the National Computer Conference to be held in Houston, Texas at about mid-June. If TI shows something that we think we could make good use of, I'll publish an extra issue, entirely dedicated to "it".

Attention weekend sailors and navigators: Robert M. Elliott, 29 Ox Hill Road, Norwich CT, 06360 USA has developed a super program called CELESTIAL RUNNING FIX. This program provides the latitude and longitude of a running fix, using any two of up to four celestial observations. No plotting is required. The program is good for any celestial body: sun, moon, planet or star, using either natural or artificial horizon.

The program uses two mag cards and the printer, but can easily be modified for calculator-only use. Robert spent about 1500 hours on the development of it over the last three years. It is "purer than Ivory soap" and it has been tested with over 100 actual computations.

Robert will send you a copy of the documentation, the instructions how to use it, worked out examples and THE PROGRAM RECORDED ON TWO MAG CARDS, all for $ 4.00. The thing is too large to fit in the TI PPC NOTES, so write Robert for a copy, please.

We had the Survival Guide for Master Library reprinted. From my experience with two earlier reprintings, they go fast. So, if you want to assure yourself a copy, send $ 10.00 to the club as soon as possible. Members abroad please add $ 2.00 for postage.

Peter Poloczek has accomplished the impossible: Combine Fast and Graphics modes in one and the same program. More on it in next issue.

Maurice E.T. Swinnen.

IN THIS ISSUE

Dear Mr. Swinnen,

Enclosed you will find what I hope to be the first if not the second program submission for the new AF-1. I hope this program will be of interest to your readers.

I will like to make some comments about the new AF-1 programmable calculator. First if you wish to get in contact with OK Instruments you use the following telephone numbers, which is not a toll free number but now it has the complete area code: (642) 774-3665 2-(886) 683-6763. Second, I did not like about the AF-1, among other things, is it's weight, too heavy, and it's size especially the size of the magnetic card, that with the increase of the postal service will make it hard on the users to submit a program.

I hope that OK Instruments will manufacture accessories such as capability to play pac-man, interface capabilities.

I wish that you could get in contact with me, my two telephone numbers are .....but remember, be careful what the number AF-1 stands for.

I think it should have been model AF-3 since it is the third time you did it to us.

Sincerely,

José M. G. Chula Vista CA.

Dear José,

I appreciate your comments on the new one. I think it is going to be a best seller.Also, many thanks for the nice 5 by 20 inch mag card you sent me. It has acquired a permanent place on the wall in my office at Walter Reed, right above my collection of (semi-dressed) Guardian Angels pictures.

Maurice.

Dear Maurice,

Please change my mailing address to ..... Also send me five (5) of the new OK# Instruments calculators and have a happy 13 33 35 24 27 21 32 32 27 36 16 13 45.

Very Truly Yours,

Tim N. San Diego, CA.

Dear Tim,

On request 1: will do.

On request 2: will do. Are you a sailor (San Diego?) and could you use an extra anchor maybe?

Su atentisimo y seguro servidor,

Maurice.

Dear Mr. Swinnen,

Your prompt response and excellent newsletter are well worthwhile the extra $ 5.00 (for first class mail) which is enclosed.

I am still in the dark about interfacing the TI-59 with the ouside world. Do you know of anyone who has done this via an RS-232 interface or IEEE control system? Surely someone must have done this in the past 5 years... Please let me know in or with your next newsletter.

By the way, what is the next programming puzzle?

Yours Sincerely,

Dave H. Canada.

Dear Dave,

I still have to receive a well-documented article on interfacing the TI-59 to an ouside device. I have a few hardware articles available, such as a TV interface to the TI-59, such that listings will appear on the creen. But nobody has told me yet if they actually built the device and if it worked satisfactorily.

Maurice.

Dear Maurice,

I received v7n3 yesterday. I liked them a lot, but cannot find any clever reason why you are spending 2 (and more) pages we pay to make the 1 April joke. Two nice program (game programs, why not having some of them) can fit much better on those pages, don't you think so? By the way, I have a TRS80 program that combines a telephone numbers to spell words. I tryed your number and get only one meaning APRIL JOKE.

I had an oportunity of playing with Sharp PC1211. I agree with your editorial (v7n3p1) complitelly. There is only one way you can use it better than TI-59: on the exams. You can put quite a nice ammount of text and formulas on it. Some of my friends pointed it.

I hope I'll receive something from you soon.

Cordially,

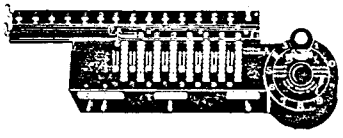
D.R. Yugoslavia.

Dear Dejan,

Be a good sport now. Man doesn't live by bread alone. Once in a while he craves for something more "tasty."

You are still young. Don't take life too seriously or you'll soon turn into a 7687 7877.

Cordially,
Maurice.



Calculatrice Fournier

BLACKJACK TUTOR,- by R^2 C^2 $SNOW^2$. This program will TEACH you how to play BLACKJACK. It will deal cards for this well known "21" game and will allow the player to determine the hands. The player is never given a hand that totals 8 or more, as this would obviously mean a HIT. The Tutor will respond with a RIGHT for the correct strategy. It will also provide the correct strategy if the player chooses the incorrect one. If you press B, the Tutor will always provide you with the correct strategy.

The algorithm this program is based on is as follows:

1. *When the dealer has a small card (2, 3, 4, 5, 6) STAND on hands of 13, 14, 15, 16.*
2. *When the dealer has a large card (7, 8, 9, 10, ace) DRAW until a count of 17 or greater has been reached.*
3. *DOUBLE DOWN when you have: a) hard 10, except when the dealer has 10 or ace, b) hard 9, except when the dealer has 7 through ace, c) hard 11, always, d) ace-2 through ace-5, when the dealer has 4, 5, 6, e) ace-6 when dealer has 2 through 6, f) ace-7 when dealer has 3 through 7.*
4. *SPLIT PAIRS: a) when you have 2s and the dealer has 3 through 7, b) when you have 3s and the dealer has 4 through 7, c) when you have 6s and the dealer has 2 through 6, d) when you have 7s and the dealer has 2 through 7, e) when you have 9s and the dealer has anything but ace-7-10, f) always when you have aces or 8s.*
5. *For ace-2 through ace-6, DRAW a card if not able to double down.*
6. *When holding ace-7: a) STAND if dealer has ace-2-7-8, b) DOUBLE DOWN if dealer has 3-4-5-6, c) DRAW if dealer has 9 or 10.*
7. *Always STAND on ace-8, ace-9.*

INSTRUCTIONS: 1. Initialize, press E, enter seed, press E'.
2. To ask for help, press B. Correct strategy will be printed.
3. To play you own strategy: to STAND, press R/S, to HIT, press A, to SPLIT PAIRS, press C, to DOUBLE DOWN, press D.

RECORDING INSTRUCTIONS: Press 7 OP 17 and key in program and data (alpha) registers 09 through 69. Press 6 OP 17 and record four card sides.
Suggestion for R69: 10 STO 69 25131526 EE 11 +/- SUM 69 INV EE. Same procedure for R60.

```
        22222.      09
      2222222.      10
   2222?22222.      11
        33300.      12
        33333.      13
        33333.      14
        33333.      15
        33333.      16
   3333333333.      17
   3333333333.      18
   3333333333.      19
           0.       20
           0.       21
       111110.      22
       111110.      23
           0.       24
     22222222.      25
           0.       26
        11111.      27
  1111111111.       28
  3311311111.       29
  3333333333.       30
  1111111110.       31
           0.       32
       22200.       33
       22200.       34
       22200.       35
       22200.       36
       22200.       37
  3000022223.       38
  3333333333.       39
  3333333333.       40
```

```
1617132717. DEALE 41
3536004133. RS UP 42
15133516.    CARD 43
45324135.    YOUR 44
1427131526. BLACK 45
2513152600. JACK  46
3637133116. STAND 47
2324370000. HIT   48
1417363700. BEST  49
14173700.    BET  50
3633272437. SPLIT 51
33132435.    PAIR 52
1632411427. DOUBL 53
16324331.    DOWN 54
1731371735. ENTER 55
36171716.    SEED 56
3524222337. RIGHT 57
3741373235. TUTOR 58
7.00000008         59
11.00000131517     60
2.00000003         61
3.00000004         62
4.00000005         63
5.00000006         64
6.00000007         65
7.00000008         66
8.00000009         67
9.00000012         68
10.000025131526    69
```

```
BLACKJACK TUTOR

 R/S     STAND
  A      HIT
  B      BEST  BET
  C      SPLIT PAIR
  D      DOUBL DOWN

         ENTER SEED

     456123.


DEALERS UP CARD  7
     YOUR CARD   8
          17.    9
   BEST STAND
DEALERS UP CARD JACK
     YOUR CARD   7
          16.    9
   BEST HIT
DEALERS UP CARD  4
     YOUR CARD   5
           9.    4
   RIGHT
DEALERS UP CARD  9
     YOUR CARD   8
          15.    7
   BEST HIT
```

BLACKJACK TUTOR, R^2 C^2 SNOW^2

000	83	GO*
001	05	05
002	76	LBL
003	18	C'
004	00	0
005	48	EXC
006	04	04
007	76	LBL
008	17	B'
009	69	OP
010	03	03
011	69	OP
012	05	05
013	69	OP
014	00	00
015	00	0
016	92	RTN
017	76	LBL
018	19	D'
019	43	RCL
020	43	43
021	42	STO
022	04	04
023	87	IFF
024	05	05
025	00	00
026	63	63
027	01	1
028	01	1
029	82	HIR
030	44	44
031	82	HIR
032	14	14
033	59	INT
034	82	HIR
035	54	54
036	85	+
037	05	5
038	09	9
039	95	=
040	42	STO
041	06	06
042	73	RC*
043	06	06
044	82	HIR
045	08	08
046	59	INT
047	92	RTN
048	76	LBL
049	16	A'
050	32	X:T
051	01	1
052	00	0
053	77	GE
054	00	00
055	58	58
056	86	STF
057	01	01
058	92	RTN
059	76	LBL
060	12	B
061	01	1
062	11	A

063	87	IFF
064	06	06
065	00	00
066	68	68
067	18	C'
068	43	RCL
069	02	02
070	91	R/S
071	61	GTO
072	00	00
073	36	36
074	86	STF
075	05	05
076	43	RCL
077	41	41
078	69	OP
079	01	01
080	43	RCL
081	42	42
082	69	OP
083	02	02
084	19	D'
085	75	-
086	01	1
087	95	=
088	22	INV
089	28	LOG
090	42	STO
091	01	01
092	18	C'
093	43	RCL
094	44	44
095	69	OP
096	02	02
097	19	D'
098	42	STO
099	02	02
100	16	A'
101	18	C'
102	86	STF
103	06	06
104	19	D'
105	67	EQ
106	02	02
107	45	45
108	44	SUM
109	02	02
110	16	A'
111	09	9
112	32	X:T
113	43	RCL
114	02	02
115	77	GE
116	01	01
117	26	26
118	86	STF
119	04	04
120	69	OP
121	05	05
122	19	D'
123	61	GTO
124	01	01
125	08	08

126	32	X:T
127	02	2
128	01	1
129	67	EQ
130	02	02
131	66	66
132	22	INV
133	87	IFF
134	04	04
135	01	01
136	45	45
137	01	1
138	01	1
139	77	GE
140	01	01
141	45	45
142	22	INV
143	86	STF
144	04	04
145	32	X:T
146	69	OP
147	06	06
148	69	OP
149	00	00
150	87	IFF
151	01	01
152	02	02
153	49	49
154	43	RCL
155	02	02
156	91	R/S
157	01	1
158	85	+
159	76	LBL
160	14	D
161	01	1
162	85	+
163	76	LBL
164	13	C
165	01	1
166	85	+
167	76	LBL
168	11	A
169	00	0
170	95	=
171	32	X:T
172	22	INV
173	87	IFF
174	04	04
175	01	01
176	81	81
177	02	2
178	04	4
179	42	STO
180	02	02
181	43	RCL
182	49	49
183	69	OP
184	02	02
185	73	RC*
186	02	02
187	55	÷

188	43	RCL
189	01	01
190	54	)
191	22	INV
192	59	INT
193	65	×
194	01	1
195	00	0
196	95	=
197	59	INT
198	67	EQ
199	02	02
200	18	18
201	32	X:T
202	00	0
203	67	EQ
204	02	02
205	25	25
206	01	1
207	67	EQ
208	02	02
209	29	29
210	02	2
211	67	EQ
212	02	02
213	37	37
214	43	RCL
215	47	47
216	17	B'
217	81	RST
218	43	RCL
219	57	57
220	69	OP
221	02	02
222	00	0
223	17	B'
224	81	RST
225	43	RCL
226	48	48
227	17	B'
228	81	RST
229	43	RCL
230	51	51
231	69	OP
232	02	02
233	43	RCL
234	52	52
235	17	B'
236	81	RST
237	43	RCL
238	53	53
239	69	OP
240	02	02
241	43	RCL
242	54	54
243	17	B'
244	81	RST
245	85	+
246	43	RCL
247	52	52
248	17	B'
249	02	2

250	00	0
251	48	EXC
252	02	02
253	44	SUM
254	02	02
255	95	=
256	22	INV
257	87	IFF
258	01	01
259	01	01
260	56	56
261	01	1
262	02	2
263	61	GTO
264	01	01
265	56	56
266	43	RCL
267	45	45
268	69	OP
269	02	02
270	43	RCL
271	46	46
272	17	B'
273	81	RST
274	76	LBL
275	15	E
276	02	2
277	85	+
278	76	LBL
279	10	E'
280	07	7
281	04	4
282	95	=
283	42	STO
284	05	05
285	07	7
286	69	OP
287	17	17
288	43	RCL
289	45	45
290	69	OP
291	01	01
292	43	RCL
293	46	46
294	69	OP
295	02	02
296	43	RCL
297	58	58
298	17	B'
299	17	B'
300	03	3
301	05	5
302	06	6
303	03	3
304	03	3
305	06	6
306	69	OP
307	01	01
308	43	RCL
309	47	47
310	17	B'
311	01	1

312	03	3
313	69	OP
314	01	01
315	43	RCL
316	48	48
317	17	B'
318	01	1
319	04	4
320	69	OP
321	01	01
322	43	RCL
323	50	50
324	69	OP
325	04	04
326	43	RCL
327	49	49
328	17	B'
329	01	1
330	05	5
331	69	OP
332	01	01
333	43	RCL
334	52	52
335	69	OP
336	04	04
337	43	RCL
338	51	51
339	17	B'
340	01	1
341	06	6
342	69	OP
343	01	01
344	43	RCL
345	54	54
346	69	OP
347	04	04
348	43	RCL
349	53	53
350	17	B'
351	17	B'
352	43	RCL
353	56	56
354	69	OP
355	04	04
356	43	RCL
357	55	55
358	17	B'
359	70	RAD
360	91	R/S
361	34	FX
362	24	CE
363	95	=
364	38	SIN
365	50	I×I
366	82	HIR
367	04	04
368	22	INV
369	59	INT
370	67	EQ
371	03	03
372	52	52
373	81	RST

IMPROVED ALPHA & NUMERIC LIST.- Lem Matteson wrote this program. He says that he used Frédéric De Mees' program from v6n2p10 to list the data in his own Loan Schedule program. It is ideal in that Frederic sure makes good use of the printing registers to get the print codes, the word and the register number into the line. Did you notice that the word has to be partially in Op 03 and OP 04 along with the register number? However, the program could be more friendly to the user. Lem never liked using R/S to enter unrelated data. How can you mark a card to show what for and when to use R/S? Lem's objective was to list all 100 registers. First he thought of entering a fake register number (another unfriendliness towards the user) but then it dawned on him that one could force banks 1 and 2 from the program to be

000	76	LBL
001	19	D'
002	53	(
003	50	I×I
004	85	+
005	01	1
006	85	+
007	34	FX
008	23	LNX
009	59	INT
010	65	×
011	02	2
012	54	)
013	92	RTN
014	76	LBL
015	18	C'
016	50	I×I
017	32	X↕T
018	01	1
019	00	0
020	00	0
021	49	PRD
022	66	66
023	32	X↕T
024	44	SUM
025	66	66
026	97	DSZ
027	64	64
028	00	00
029	41	41
030	05	5
031	42	STO
032	64	64
033	25	CLR
034	48	EXC
035	66	66
036	84	OP*
037	65	65
038	01	1
039	44	SUM
040	65	65
041	92	RTN
042	76	LBL
043	14	D
044	32	X↕T
045	07	7
046	69	OP
047	17	17
048	32	X↕T
049	82	HIR
050	04	04
051	75	-
052	06	6
053	00	0
054	95	=
055	61	GTO
056	00	00
057	76	76
058	76	LBL
059	15	E
060	75	-
061	05	5

062	09	9
063	95	=
064	61	GTO
065	00	00
066	88	88
067	76	LBL
068	11	A
069	32	X↕T
070	07	7
071	69	OP
072	17	17
073	32	X↕T
074	82	HIR
075	04	04
076	42	STO
077	68	68
078	42	STO
079	69	69
080	82	HIR
081	14	14
082	91	R/S
083	76	LBL
084	12	B
085	85	+
086	01	1
087	95	=
088	22	INV
089	44	SUM
090	69	69
091	82	HIR
092	14	14
093	55	÷
094	01	1
095	00	0
096	75	-
097	69	OP
098	00	00
099	42	STO
100	67	67
101	19	D'
102	52	EE
103	02	2
104	48	EXC
105	67	67
106	95	=
107	65	×
108	01	1
109	00	0
110	95	=
111	19	D'
112	44	SUM
113	67	67
114	43	RCL
115	67	67
116	22	INV
117	52	EE
118	69	OP
119	04	04
120	73	RC*
121	68	68
122	42	STO

123	67	67
124	50	I×I
125	32	X↕T
126	00	0
127	67	EQ
128	03	03
129	41	41
130	01	1
131	52	EE
132	01	1
133	03	3
134	22	INV
135	77	GE
136	02	02
137	70	70
138	01	1
139	52	EE
140	03	3
141	94	+/-
142	77	GE
143	02	02
144	70	70
145	05	5
146	42	STO
147	64	64
148	01	1
149	42	STO
150	65	65
151	25	CLR
152	42	STO
153	66	66
154	73	RC*
155	68	68
156	29	CP
157	77	GE
158	01	01
159	63	63
160	02	2
161	00	0
162	18	C'
163	22	INV
164	86	STF
165	05	05
166	43	RCL
167	67	67
168	52	EE
169	55	÷
170	52	EE
171	00	0
172	00	0
173	95	=
174	28	LOG
175	58	FIX
176	00	00
177	22	INV
178	52	EE
179	52	EE
180	58	FIX
181	09	09
182	29	CP
183	77	GE

184	01	01
185	88	88
186	01	1
187	94	+/-
188	82	HIR
189	03	03
190	87	IFF
191	05	05
192	02	02
193	08	08
194	29	CP
195	82	HIR
196	13	13
197	77	GE
198	02	02
199	08	08
200	86	STF
201	05	05
202	04	4
203	00	0
204	18	C'
205	61	GTO
206	02	02
207	39	39
208	43	RCL
209	67	67
210	55	÷
211	82	HIR
212	13	13
213	22	INV
214	28	LOG
215	52	EE
216	75	-
217	59	INT
218	82	HIR
219	02	02
220	95	=
221	65	×
222	82	HIR
223	13	13
224	22	INV
225	28	LOG
226	52	EE
227	95	=
228	42	STO
229	67	67
230	01	1
231	82	HIR
232	53	53
233	82	HIR
234	12	12
235	22	INV
236	52	EE
237	19	D'
238	18	C'
239	29	CP
240	43	RCL
241	67	67
242	22	INV
243	67	EQ
244	01	01

245	90	90
246	22	INV
247	87	IFF
248	05	05
249	01	01
250	90	90
251	05	5
252	32	X↕T
253	43	RCL
254	64	64
255	67	EQ
256	02	02
257	87	87
258	04	4
259	32	X↕T
260	43	RCL
261	65	65
262	67	EQ
263	02	02
264	87	87
265	00	0
266	18	C'
267	61	GTO
268	02	02
269	51	51
270	32	X↕T
271	75	-
272	52	EE
273	95	=
274	29	CP
275	67	EQ
276	02	02
277	80	80
278	25	CLR
279	35	1/X
280	73	RC*
281	68	68
282	69	OP
283	06	06
284	61	GTO
285	03	03
286	41	41
287	01	1
288	82	HIR
289	37	37
290	82	HIR
291	38	38
292	87	IFF
293	00	00
294	03	03
295	39	39
296	01	1
297	00	0
298	22	INV
299	28	LOG
300	32	X↕T
301	73	RC*
302	68	68
303	50	I×I
304	77	GE
305	03	03

306	39	39
307	29	CP
308	22	INV
309	59	INT
310	22	INV
311	67	EQ
312	03	03
313	39	39
314	73	RC*
315	68	68
316	50	I×I
317	55	÷
318	01	1
319	52	EE
320	01	1
321	06	6
322	65	×
323	82	HIR
324	37	37
325	01	1
326	52	EE
327	01	1
328	02	2
329	95	=
330	22	INV
331	59	INT
332	55	÷
333	01	1
334	00	0
335	00	0
336	95	=
337	82	HIR
338	38	38
339	69	OP
340	05	05
341	01	1
342	44	SUM
343	68	68
344	82	HIR
345	34	34
346	25	CLR
347	97	DSZ
348	69	69
349	00	00
350	91	91
351	06	6
352	69	OP
353	17	17
354	25	CLR
355	92	RTN
356	76	LBL
357	16	A'
358	86	STF
359	00	00
360	91	R/S
361	76	LBL
362	17	B'
363	22	INV
364	86	STF
365	00	00
366	91	R/S

listed into banks 3 and 4 and let the listing program add 60 to it! So, his user instructions say: Enter the low register number and press A. Enter the high register number and press B. To list press C. The foregoing only when you want registers 00 through 59. Now when you want to list registers 60 through 99 you first of all force those banks into banks 1 and 2 by means of -1 and -2. You then enter the correct low register and press D, enter the correct high register and press E, after which C will start your list. In the example on the right you see in the left column data listed from registers 05 through 20, on the top listed by means of INV LIST and on the bottom by means of this program.

```
 5151513524.        05          2332430015.        00
 4217350015.        06          3241271600.        01
 3532363624.        07          4532410016.        02
 3122515151.        08          3200000000.        03
 1331004700.        09          3700230013.        04
 4323320071.        10            37007100.        05
   30133100.        11          1532312235.        06
   21324400.        12          1337365700.        07
   23173100.        13          4532410016.        08
 1532353100.        14          2416002437.        09
 6255555555.        15          4335323122.        10
 5555555562.        16            17313735.        11
     370100.        17          4500737373.        12
 3001421700.        18          5555555555.        13
 3335173636.        19
 6200000000.        20

 5151513524. ***RI  05          2332430015. HOW C  60
 4217350015. VER C  06          3241271600. OULD   61
 3532363624. ROSSI  07          4532410016. YOU D  62
 3122515151. NG***  08          3200000000. D      63
 1331004700. AN +   09          3700230013. T H A  64
 4323320071. WHO ?  10          37007100.     T ?  65
 30133100.    MAN   11          1532312235. CONGR  66
 21324400.    FOX   12          1337365700. ATS,   67
 23173100.    HEN   13          4532410016. YOU D  68
 1532353100. CORN   14          2416002437. ID IT  69
 6255555555. ;(((((  15         4335323122. WRONG  70
 5555555562. (((((;  16           17313735.   ENTR  71
 370100.      TO    17          4500737373. Y ??? 72
 3001421700. MOVE   18          5555555555. (((((  73
 3335173636. PRESS  19
 6200000000. ;      20
```

Now in the right hand column you see data which were originally in registers 60 through 73, forced into bank 1 and listed by means of INV LIST. On the bottom the same data listed by means of this program. Note the correct register numbers, achieved by entering 60 into D and 73 into E, after which C was pressed to start the listing.

FAST MODE and GRAPHICS MODE IN THE SAME PROGRAM? Peter Poloczek of Frankfurt, West Germany accomplished this feat recently and showed a method how it could be done. It requires rather a lot of button pushing, but, with time, that could be reduced to an absolute minimum and, who knows, maybe even made transparent to the user. Dejan Ristanović, who figures rather prominently in this issue,has used this technique to write a fascinating program called FAST SCRIBBLER. It permits you to draw pictures such as Frédéric De Mees' Mickey Mouse picture in Graphics Mode at the increased speed of Fast Mode. And, using Richard Snow's technique in the Stars and Stripes program, it prints out the required buttons to be pushed by means of a programmed LIST.

If I still have enough space left over in this issue, I will publish Dejan's FAST SCRIBBLER. Otherwise, it will appear in next issue.

SUPERCKECKSUM. It is my intention to use one of the many versions in existence to enable verification of correct keying-in of programs, by publishing the checksum with each published program, just as PPX is doing. Currently there is an evaluation of the many versions of checksum programs going on among reviewers in the US, in Denmark and in Sweden. Elmer Clausen enhanced Jules Bell's program quite a lot, while Lars Hedlund rewrote it in Fast Mode. That last one is a joy to use and seems to be the winner up to now. But then, Hans Peter Nielsen, the editor of PGM in Denmark says that it may be possible to even improve on the algorithm Jules used. Robert Prins in Holland also casts doubt on Jules' statement that the algorithm is absolutely fail-safe. I will keep you informed.

RANDOM NUMBERS,- Bertrand N. Bauer of Parameter Investigation, 2423 Ridgeway Ave., Evanston, IL, USA, writes in Keyboard, Sept-Oct., 1981(*Keyboard is an HP Journal for users of the 9825 desktop calculator, one of the few at HP that do not use RPN but rather the AOS system)* about a method of producing random numbers good enough for professional use. He bases his findings on an algorithm developed by Knuth in Donald E. Knuth, Seminumerical Algorithms, Vol.2, The Art of Computer Programming, Reading MA, Addison-Wesley, 2nd printing with revisions, 1971, pp. 82-97.

The algorithms passes from one 11-digit random fraction to the next by multiplying the current random fraction by 31 622 777 021, then adding 10^{-11} and finally retaining only the fractional portion of the result. Or, as the mathematicians would write:

$$R' = \text{fra}((31\ 622\ 777\ 021)(R) + 10^{-11})$$

This algorithm has a lot of desirable characteristics: It generates every possible 11-digit fraction before it repeats itself. Thus, the "period" of this generator is 10^{11} or 100 billion, that is the number of values of R' before repetition occurs. And this period is always 100 billion, no matter what the starting seed is.

Knuth also talks about a randomness test in the reference above. He calls it a spectral test and the result of it are three values: C2, C3 and C4, which will determine if a certain algorithm to produce random numbers is good enough for specific categories of use. It so happens that the above algorithm has a C2=1.22, a C3=1.10 and a C4=3.84, which Knuth categorizes as good enough for profesional use.

The limitations are that, if the seed is the same for each different run, the same series of random numbers will be produced. This might be desirable if you want to test a specific program, to compare results over several runs. If a randomized series of random numbers has to be produced, a "random" seed has to be entered each time to start a new run. On the TI58/59 this can be accomplished rather easily by writing at 000:

LBL E OP 29 RST ; press E and go drink a cup of coffee. When you come back, press R/S and RCL 09. You will find a suitably random number in R09. To convert it to a fraction, divide it by 1 EE 11.

From the above you might have gleaned a serious shortcoming of the TI-59: it should be possible to multiply two 11-digit numbers in order to execute the algorithm. Direct multiplication is, of course, not possible, but we can use "double precision arithmatic" to get to the required accuracy. The idea behind this is to write the 11-digit number as the sum of a 5-significant-digit number and a 6-significant-digit number, thus 31 622 (10^6) + 777 021. Then do the same to the random fraction. Next we apply the algebraic theorem (a+b)(c+d) = ac + ad + bc + bd. Individually, each of these four products has 12 significant digits maximum. We need only retain the digits to the right of the decimal point to obtain the desired accuracy. It turns out that the product (ac) never has any such digits, so we may discard this term.

So, now we may write a program according to the following procedure:

Step 1: $Y = (\text{int}((R)(10^5)))(10^{-5})$
Step 2: $Z = R - Y$
Step 3: $U = \text{fra}(31\ 622)(10^6)(Z)$
Step 4: $V = \text{fra}((777\ 021)(Y))$
Step 5: $W = \text{fra}((777\ 021)(Z))$
Step 6: $R' = \text{fra}(U + V + W + 10^{-11})$

Test data from a seed of .500 000 000 01 yield Y=.5; Z=.000 000 000 01; U=.316 22; V=.5; W=.000 007 770 21 and R'=.816 227 770 22;

With the program I have included a few more test data, which were verified on the 9825. As a 13-digit output I used Charlie Williamson's 13-digit printer from v7n4/5p11. I accessed it by leaving off the last R/S of my program and writing GTO NNN instead. The step NNN is the first step of his LBL A. Admittedly, you will have to wait 33 sec to print the final 13-digit output, but it was only done to have a nice print out. If you adopt this random generator program for one of your programs, it will, of course, not be necessary to also include a 13-digit printer.

Also, the PRT statements to print out Y, Z, U, V, and W can be omitted and are used here only as debugging aids.

Maurice E.T. Swinnen

Random Numbers, M.Swinnen, listing.

```
            1.-11
            0.
            1.-11
       0.31622
            0.
  .0000077702
  3.162277702200 E-01

  .3162277702
       0.31622
  .0000077702
       0.89684
       0.58062
  .0376241146
  5.150841146300 E-01

  .5150841146
       0.51508
  .0000041146
       0.82986
       0.97668
  .1971539172
  3.693917240000 E-03
```

```
000  76 LBL
001  11  A
002  99 PRT
003  42 STO
004  00  00
005  65  ×
006  01  1
007  52 EE
008  05  5
009  95  =
010  59 INT
011  65  ×
012  01  1
013  52 EE
014  05  5
015  94 +/-
016  95  =
017  22 INV
018  52 EE
019  42 STO
020  01  01
021  99 PRT
022  75  -
023  43 RCL
```

```
024  00  00
025  95  =
026  94 +/-
027  42 STO
028  02  02
029  99 PRT
030  65  ×
031  01  1
032  52 EE
033  06  6
034  65  ×
035  03  3
036  01  1
037  06  6
038  02  2
039  02  2
040  95  =
041  22 INV
042  59 INT
043  22 INV
044  52 EE
045  99 PRT
046  42 STO
047  03  03
```

```
048  43 RCL
049  01  01
050  65  ×
051  07  7
052  07  7
053  07  7
054  00  0
055  02  2
056  01  1
057  42 STO
058  04  04
059  95  =
060  22 INV
061  59 INT
062  99 PRT
063  42 STO
064  05  05
065  43 RCL
066  02  02
067  65  ×
068  43 RCL
069  04  04
070  95  =
071  22 INV
```

```
072  59 INT
073  99 PRT
074  85  +
075  43 RCL
076  05  05
077  85  +
078  43 RCL
079  03  03
080  85  +
081  01  1
082  52 EE
083  01  1
084  01  1
085  94 +/-
086  95  =
087  22 INV
088  52 EE
089  22 INV
090  59 INT
091  42 STO
092  00  00
093  99 PRT
094  91 R/S
```

HEX CODES,-Dejan Ristanović in Belgrade, Yugoslavia has been experimenting with different hex codes on step 16, this after the usual initilization PGM12 SBR 444 R/S... The experments were done with the printer attached. Without the printer they may behave differently. Also, Dejan complains that most of the hex codes work differently in Trace mode versus Normal mode.

In the following list the character or digits in parenthesis following the hex codes is the code of the instruction to be put at step 16.

h62 (A) in Trace list as OSS* and does some strange things. The next two digits make up the address of a register.

h82 (31) lists as *GT

h73 (22) lists as *ST. The next two digits are the address of a register. 12 Digits of that register, the integer part, are printed. This might by usefull sometimes with the calculator used manually. Unfortunately, it works only in Trace.

h93 (42) Has some meaning I cannot discover. It appears that this instruction needs a five-digit address. When you SST through it, it acts as a decimal point. Maybe it just hardens the display? The next five instructions are ingnored if they are numbers, the same way as when you SST through a normal GTO instruction.

h00 acts as List.

h64 (13) acts as ABS(X) in Trace. Otherwise, it is all clear. It can be used as a Trace detector, but with fatal consequences.

h84 (33) acts as RAD in Trace. Otherwise it acts as a strange RCL.

h94 (43) enters EE 12. In Trace it acts as GRAD.

h53 (2) is all clear. In Trace it is EE.

h44 (93) acts as ln(0). In Trace it is TAN. This can be used as a nice trick to detect if the printer is in Trace or not. The only thing I cannot find is the exact way as HOW to use it.

h75 (24) is DEG in Trace. Otherwise it acts as some kind of RCL. A Trace detector is simply h75 16. It jumps to LBL A' in Trace mode only.

h85 (34) is RAD in Trace. Otherwise it also is a strange RCL.

It is a pity we don't have a Dr. W. Wickes in our club, *(Dr. Wickes wrote THE book on synthetic programming for the HP-41C.Ed.)* to write one titled "Synthetic Programming on the TI-59. After I read Dr.Wickes' book, the HP41C had no secrets for me anymore."

PERIODIC TABLE,- Lem Matteson of Kansas City, MO is the author of that program, which was published in v7n4/5p18 and 19. The listing was done in the usual way, you know, TI-59 cum PC100A. Only Lem has this habit of blotting out unused zeros in GTO direct addres instructions by means of a black pencil. Rather that trying to remove his pencil marks, I just used printers' white-out. This to satisfy numerous inquiries as to how those zeros had disappeared.

A few members told me that the program contained some typos, nothing big, though. Greg Hoen says that step 130 should be 4 instead of 5, because the author wanted to set flag 3 for numbers 1 to 4. With 5 there in step 130 the program sets flag 3 for numbers 1 through 5.

Additionally, step 141 should be 3 instead of 4. Also, atomic number 99 prints as 0. Greg Hoen says that his chemistry reference material is ancient and that there could have easily been a name change. But it should be ES for Einsteinium and register 45 then should be -1736.254323288 vs -3256.2543289.

PROGRAM PROTECTION,- Over the years I have seen many card protection schemes. The most obvious one is to record your card with a -1 through -4. However, the protection flag can be bypassed by a variety of methods and the protection can be rather easily broken. Other schemes I have seen entail a password, a secret number that has to be matched to get access to the program. Simply listing the program will reveal, of course, the password. Therefore, in my opion, it doesn't pay to try to protect a mag card.

On the other hand, it is rather simple to protect a CROM or module. Dave Leising sent me a few schemes he proposes:

1. Making any program longer than 960 steps, if needed by means of fake NOPs, will prevent it from being downloaded by means of OP 09. Very effective protection of individual programs in a module.

2. The last program in a CROM can be assigned a fictitiously high upper address in the internal CROM map, such that it cannot be downloaded, even though the program contains less than 960 steps.

3. Unauthorized execution of a protected module (CROM) can be prevented rather easily by means of the password scheme. Suppose the password is 93716. To get access to the entire CROM one would have to punch in: 93716 2nd PGM 15 A.

And at PGM 15 in the module we would have:

LBL A HIR 03 E'..... Now at LBL E' 93716 X:T HIR 13 EQ NNN 2nd SIN RTN

At NNN the program would continue if the correct password was entered. Otherwise the calculator would encounter the famous "2nd SIN" sequence and go to never-never land, from which a return is only possible by turning off the calculator.

E' calls could be distributed strategically along the entire module, preventing analysis by absolute address entry attempts.

The "2nd SIN" crash doesn't work on the TI-58, but I am sure there is a crash sequence that works on that calculator too.

HOW TO CHANGE YOUR TI-57 INTO A TI-57C. Dejan Ristanović tells me that the French magazine "L'Ordinateur de Poche" says how to do it, in its No.4 issue of 1982. At the end of your program, step 48, place the following: 2nd EXC SST LBL 1. Press LRN. Now, when you want to switch your calculator off, to save the memory, press GTO 2nd 48 R/S INV STO 3 +/- +/- . The display will go blank. Thus, the machine will not supply any more power to light the LED display. It uses even less energy than the TI-58C to save both data and program.
When you want to switch the calculator back on, press INV FIX CLR.

52-NOTES,- Richard Vanderburgh, 9459 Taylorsville Road, Huber Heights OH, 45424, USA, the editor of that newsletter, now defunct, sends me the following information: 34 Back issues of the 52-Notes are still available at $ 1.50 each, $ 2.00 if abroad, US funds. The first twelve issues cover mainly SR-52, SR-56 and PC100. The remaining 22 issues cover mainly TI-58, TI-59 and PC100C.

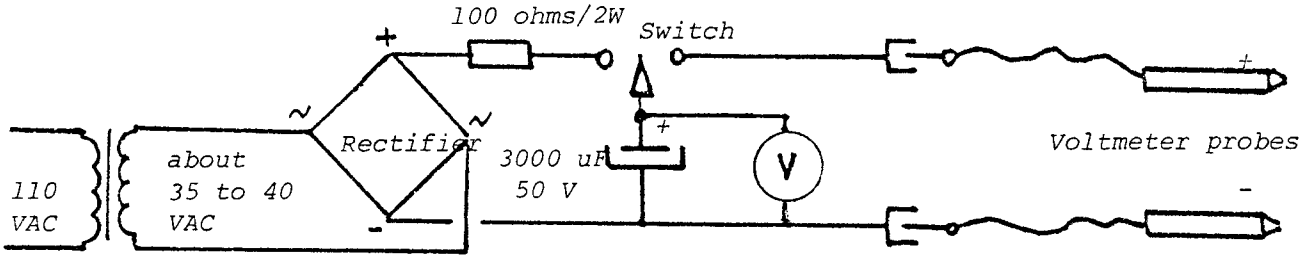
Note mainly the renaming of the "city" from "Dayton" to "Huber Heights" and the changed prices for each issue.

EMERGENCY POWER, Palmer Hanson Jr. writes:" In "Battery Power in an Emergency" (v6n8p14)
---------- Bob Fruit discussed the use of a six volt lantern pack as an emergency power source for the TI-59. A more likely operation of the TI-59 with the power source outside the normal operating range occurs when one of the three cells of the BP-1A battery pack develops an internal short. Such a malfunction is typical of the nickel cadmium cells used in the battery pack. The result is that no amount of charging with either the model AC 9131 or with the charging circuit of the PC100 will eliminate the short. (The Zapper, described below,will. Ed.) *The output of a "fully" charged battery pack becomes somewhat less than 3 volts. And with this low power output:*

a. When running with a battery pack only, the display will be somewhat dim. Calculations will be performed satisfactorily. Reading and writing of magnetic cards will be erratic at best, and often impossible.Unpredictable displays result when trying to read a card, for example, flashing two-digit numbers.
b. With the defective battery pack installed in the PC100 charging receptacle, all reactions will be normal with the calculator installed on the PC100.
c. With the defective battery pack installed and with the AC-9131 adaptor connected, the reactions will be the same as with the defective battery pack only.

Since originally indentifying the above malfunction I have helped several others with calculators which would read and write the magnetic cards erratically. Could this type of malfunction have been the source of "hundreds upon hundreds of misreads of my cards.." as reported in the "Mailbag" of v6n6/7p2 ? "

THE ZAPPER, Maurice E.T. Swinnen. In connection with the above, especially with the words
---------- " when one of the three cells of the battery pack develops an internal short.." I had been annoyed several times, enough to start a "course of inquiry" as to the source of this annoyance. Dissecting a defective battery I found a well defined dendrite in its



insides. While the dendrite proved to be conductive, it definitely was identified as the villain who shorted my battery. A simple application of about 50 volts evaporated said dendrite. Thusly, the Zapper was born.

The schematic diagram above shows what is needed. A power supply, either one you put together from a discarded transformer and a bridge rectifier or an existing one, capable of supplying about 50 volts at a few milliamps will do. An electrolytic capacitor of at least 3000 uF at 50 volts has enough energy stored to zap even the most recalcitrant battery. A voltmeter (your VOM will do nicely if it has in the vicinity of 100,000 ohms per volt internal impedance so as not to load too heavily the charged capacitor) is handy to find out when your capacitor is charged up and ready for the next zap. A switch allows you to charge the capacitor slowly when switched to the left and to discharge the capacitor quickly across the NiCad cell when turned to the right. To this end, drill three small holes, about 2 mm, on either side of the battery pack, THROUGH THE PLASTIC ONLY, so as to expose the heads (metal) of each cell. Your voltmeter probes come in handy to apply "zapping power" through these small holes to each cell. Sometimes, a couple of applications are needed. It is good practice to zap a known shorted cell a couple of times, then to charge it for an hour or so in the PC100. If it still doesn't take a charge, zap again.

I have had success in about 90 % of all the cases, taken over more than forty batteries zapped, TI, HP and others.

PATENTS: A few more calculator-related patents have been received:

3,924,110 CALCULATOR SYSTEM FEATURING A SUBROUTINE REGISTER, Filed Dept.13, 1973. Awarded Dec. 2, 1975 to Michael J. Cochran & Charles P. Grant.

3,916,169 CALCULATOR SYSTEM HAVING PRECHARGED VIRTUAL GROUND MEMORY, Filed Sept. 13, 1973. Awarded Oct. 28, 1975 to Michael J. Cochran & Charles P. Grant.

3,919,532 CALCULATOR SYSTEM HAVING AN EXCHANGE DATA MEMORY REGISTER, Filed Sept. 13, 1973. Awarded Nov. 11, 1975 to Michael J. Cochran & Charles P. Grant.

3,919,536 PRECHARGED DIGITAL ADDER AND CARRY CIRCUIT, Filed Sept. 13, 1973. Awarded Nov. 11, 1975 to Michael J. Cochran & Charles P. Grant.

3,922,538 CALCULATOR SYSTEM FEATURING RELATIVE PROGRAM MEMORY. Filed Sept. 13, 1973. Awarded Nov. 25, 1975 to Michael J. Cochran and Charles P. Grant.

It is curious to note, with respect to the last patent, that TI never used a relative memory in any of its calculators. But HP, on the other hand, has it incorporated in many of its programmables. Could it be that the first designs for the HP calculators were done at TI? I know that the very first ones were manufactured at TI.

For our newcomers: A relative memory means that you might have a command such as GTO RELATIVE. That would mean you would advance so many steps as the contents in a specific register would indicate. That would work from any program step. It would just as well work with a positive as with a negative number, meaning you would either go forward or backward relative to your present position. It is very easy to insert or delete steps in a RELATIVE ADDRESS mode as opposed to a DIRECT ADDRESS mode.

PUZZLE,- Harald M. Otto of West Germany offers the following puzzle:
"Devise a routine that will invert the digits of a positive whole number comprising between 2 and 13 digits." Of course, only 10 digits will be shown in the display, but we now have enough routines to even print a 13-digit number and check if the reversing routine works correctly.

To make it absolutely unambiguous as to what is wanted, here are a few samples: 123456789 has to become 987654321 and 1122334455667 should become 7665544332211.

FLASHTUBE CIRCUIT PROGRAM,- Eamon Murphy, Electro-Optical System Design, April 1982, pp. 41-45.

This TI-59/PC100 program will work primarily for linear flashtubes. It gives a simple design procedure to optimize single-mesh driving circuits. The program may be tuned to a particular manufacturer's tube by adjusting two values in the program.

All design equations are given plus a sample calculation. Author's address is Photodyne Inc. 948 Tourmaline Drive, Newbury Park, CA, 91320, USA.

PATENTS,- Still more patents arrived:

3,904,862 CALCULATOR SYSTEM HAVING A CONSTANT MEMORY, Filed Sept.13, 1973, awarded Sept.9, 1975 to Michael J. Cochran & Charles P. Grant.

3,904,863 CALCULATOR SYSTEM USING INSTRUCTION WORDS AS DATA, Filed Sept. 13, 1973, awarded Sept.9, 1975 to Michael J. Cochran & Charles P. Grant.

MODULE SELECTOR,- When ordering the module selector from Systems 7 in Houston TX at the greatly reduced price, as announced in v7n4/5p9, please be adviced that this firm requires an extra $ 4.00 US as handling and shipping charges per module switcher. Even at that, it is still considerably cheaper than directly from American Microproducts.

PROGRAMMING THE TI-55, by Stephen L. Snover and Mark A. Spikell, Prentice-Hall 1982,
---------- 116 pages, $ 7.95 US.

Although I couldn't imagine why anyone would ever buy a TI-55 (*) when for a few dollars more you can buy the so much more powerful TI-57, this book is well-written and could be a real help to, say, the high school student who just bought one of these programmables and finds out that the manual itself doesn't provide all of the answers.

Professors Stover and Spikell, old hands at this game, have milked the last drop out of this one too and have explained some of the function into such detail that it made me infortable. Could anyone be that dense to require such detail? But then, they are the educators, not I, and they probably know what they are doing.

The book contains a wealth of worked-out examples and procedures to teach somebody what programming is all about. Because the TI-55 lacks a decision-making test, the authors even devised one, based on the fact that if you count down a register in steps of one and use the contents each time as the divisor in a 1/x test, at zero it becomes undefined and the calculator stops with a flashing 999999. This sort of poor-man's DSZ (or should it rather be "decrement and stop on zero"?) is neat. It allows you to program finite loops now.

(*) To justify my rather sour remark about the TI-55: it doesn't have decision-making test, nor does it have labels or even editing capabilities. Yes, if you forget to key in a step, you have to do it all over again. As a consolation, no program can be longer than 32 steps. A redeeming feature of the 55 is, that it has more special functions than any other calculator: conversions, statistics, even sinh, cosh and tanh.

If the TI-55 had come along in, say, 1972, it would have been a sensation. As it is, this is 1982 and we are getting jaded with respect to programmables. In my opinion, the TI-57 impacts rather heavily on the TI-55, with more programming capability at a slightly increased price.

HOW WELL DOES THAT LINE FIT, by George Wm. Thomson, Detroit, Michigan.
---------- The built-in hard-wired functions of the TI-59 make the fitting of straight lines an easy task. It is assumed that the X-values are free from error and that the data points are of equal weight. The Method of Least Squares is used to minimize the sum of the squares of the deviations of the observed values from the calculated line. The only measure of "goodness of fit" of the points which is provided is the Correlation Coefficient (OP 13) which is a very poor kind of measure for most engineering and scientific work. Fortunately the much more useful Standard Error of FIT, sometimes called the Mean Square Error of Fit, can be obtained from the short subroutine below:

LBL A 1 - OP 13 X^2 = X OP 11 X RCL 03 = DIV (RCL 03 - 2 = $\sqrt{x}$ PRT RTN

The result is in the same units as the Y-values with n-2 degrees of freedom. The SE FIT is obtained by adding up the squares of the deviations of the calculated values of Y from the input values, dividing the sum by n-2 and taking the square root. Since none of the stored data in registers 01 through 06 is altered either by OP 12 to 15 or the above subroutine, adding or dropping data points from a set is easily done without reentering the whole set.
To use the subroutine, press A after at least three data points have been entered. The SE FIT will be printed. Points can then be added using the X:T and Σ+ keys. The subroutine permits evaluation of the set of data at any stage.
This subroutine is almost identical with a routine I derived for the SR-51-II several years ago.

GAUSS REDUCTION OF MATRICES M·N,- Saša Nick and Dejan Ristanović, the "Worthington &
---------- Regelman"'s of Belgrade, Yugoslavia have done it again. This time it is a mathematical program of class A-1. The program can solve any (and we mean ANY) system of m linear equations with n unknowns (m·(n+1) < 85). If you are familiar with linear algebra, you can use it to find the kernel (ker f) and the image (Im f) of any linear transformation, and so on. The program is very short (only 240 steps), will therefore fit on one side of a mag card, and will accomodate the unusual number of m linear equations with n unknowns of (m·(n+1) = 84. The program could be made more user-friendly, but that would entail adding about 80 steps, which would reduce the size of the systems it being able to solve. The authors claim that they would be able to write such a "user-friendly" program, if members' interest warrants it.

The system is generally expressed as:

$$a_{11}x_1 + a_{12}x_2 + \ldots\ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots\ldots + a_{2n}x_n = b_2$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots\ldots + a_{mn}x_n = b_m$$

To solve such a system, you have to write it as a matrix m·(n+1). You now simply enter the coefficients of the matrix and start the program. After a short interval of computation you will receive another matrix that looks like:

$$\begin{Vmatrix} a_{11} & a_{12} & \ldots\ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots\ldots & a_{2n} & b_2 \\ a_{m1} & a_{m2} & \ldots\ldots & a_{mn} & b_m \end{Vmatrix} \sim \begin{Vmatrix} 1 & a'_{12} & a'_{13} & \ldots\ldots & a'_{1n} & b'_1 \\ 0 & 1 & a'_{23} & \ldots\ldots & a'_{2n} & b'_2 \\ 0 & 0 & 0 & \ldots\ldots & a'_{mn} & b'_m \end{Vmatrix}$$

Once you have the second matrix it will be easy to solve the system. If there are more unknowns than equations, don't expect to find their exact values: some of the unknowns will be a function of other unknowns. If there are more aquations than there are unknowns it is not entirely sure that the system has a solution at all. To check on this, you might solve the m·n system and manually check if the other equations (there are m-n of them) are good. You can also use this program by imagining more unknowns (m-n new ones) and entering their coefficients as zero. Then reduce the matrx and find out if there is a line that looks like 0xj=5. If there is such a line, the system doesn't have a solution.

User instructions: Key in the program and save it one one side of a mag card in 6 OP 17 partitioning. To use it, enter n+1 (number of columns) and press X:T. Enter the number of rows (m) and press A. Enter a_{11} and press R/S. Enter a_{12}...a_{1n} and press R/S each time. Continue with this until you have entered all the coefficients.

Press B and wait for the computation to finish. The display will show a'_{11}. Press R/S and the display will show a'_{12} and so on. It is very important to realize that the calculator will NOT display all the elements. It will only display the triangle matrix. Thus, after a'_{1n} and b'_1 the display will show a_{22}. (a_{21} is always zero. The knowledge of this will save you some time, obviously)

For a new case, enter (n+1), press X:T, and so on.

Operating limits and warnings: The calculator MUST NOT be in the ENG mode. You are responsible for checking if the number of rows and columns is alright. Otherwise, the calculator will flash after 85 coefficients have been entered.

The program cannot solve systems with some of the coefficients very large, say 10000000000 while others very small, say 0.0000000001. Also if all of the coefficients are too large (or too small) the authors suggest you transform the equations manually before you use the program.

See program on next page, please.

THIRTEEN DIGIT MODULO 30 SPEEDY FACTOR FINDER.- Palmer O. Hanson Jr. has completed the do-
---------- cumentation on it and submitted it to PPX. It uses STF-h12 methods to enter into Fast Mode. Write to Texas Instruments Inc. PPX, P.O. BOX 53, Lubbock TX 79408 for a copy ($ 4.00 + $ 2.00 handling + tax) of this 17-page program.

GAUSS REDUCTION OF MATRICES M·N, Saša Nick & Dejan Ristanović.

```
000  76 LBL
001  10 E'
002  42 STO
003  03  03
004  75  -
005  01  1
006  95  =
007  65  ×
008  82 HIR
009  17  17
010  85  +
011  43 RCL
012  04  04
013  85  +
014  05  5
015  95  =
016  42 STO
017  05  05
018  92 RTN
019  76 LBL
020  19 D'
021  82 HIR
022  17  17
023  42 STO
024  04  04
025  75  -
026  43 RCL
027  00  00
028  95  =
029  42 STO
030  02  02
031  69 OP
032  22  22
033  92 RTN
034  76 LBL
035  12  B
036  58 FIX
037  08  08
038  29 CP
039  69 OP

040  20  20
041  43 RCL
042  00  00
043  42 STO
044  01  01
045  42 STO
046  04  04
047  10 E'
048  25 CLR
049  73 RC*
050  05  05
051  52 EE
052  22 INV
053  67 EQ
054  01  01
055  08  08
056  29 CP
057  25 CLR
058  73 RC*
059  05  05
060  52 EE
061  22 INV
062  67 EQ
063  00  00
064  82  82
065  82 HIR
066  17  17
067  44 SUM
068  05  05
069  69 OP
070  21  21
071  43 RCL
072  01  01
073  32 X:T
074  82 HIR
075  18  18
076  77 GE
077  00  00
078  56  56
079  61 GTO

080  01  01
081  80  80
082  19 D'
083  43 RCL
084  00  00
085  10 E'
086  73 RC*
087  05  05
088  32 X:T
089  43 RCL
090  01  01
091  10 E'
092  32 X:T
093  63 EX*
094  05  05
095  32 X:T
096  43 RCL
097  00  00
098  10 E'
099  32 X:T
100  72 ST*
101  05  05
102  69 OP
103  34  34
104  97 DSZ
105  02  02
106  00  00
107  83  83
108  43 RCL
109  00  00
110  42 STO
111  04  04
112  10 E'
113  73 RC*
114  05  05
115  42 STO
116  01  01
117  19 D'
118  43 RCL
119  03  03

120  10 E'
121  43 RCL
122  01  01
123  22 INV
124  64 PD*
125  05  05
126  69 OP
127  35  35
128  97 DSZ
129  02  02
130  01  01
131  21  21
132  82 HIR
133  18  18
134  42 STO
135  01  01
136  43 RCL
137  01  01
138  32 X:T
139  43 RCL
140  00  00
141  67 EQ
142  01  01
143  76  76
144  42 STO
145  04  04
146  32 X:T
147  10 E'
148  73 RC*
149  05  05
150  82 HIR
151  06  06
152  19 D'
153  43 RCL
154  00  00
155  10 E'
156  73 RC*
157  05  05
158  65  ×
159  82 HIR

160  16  16
161  95  =
162  32 X:T
163  43 RCL
164  01  01
165  10 E'
166  32 X:T
167  22 INV
168  74 SM*
169  05  05
170  69 OP
171  34  34
172  97 DSZ
173  02  02
174  01  01
175  53  53
176  97 DSZ
177  01  01
178  01  01
179  36  36
180  82 HIR
181  18  18
182  32 X:T
183  43 RCL
184  00  00
185  22 INV
186  77 GE
187  12 B
188  22 INV
189  58 FIX
190  06  6
191  42 STO
192  01  01
193  25 CLR
194  73 RC*
195  01  01
196  91 R/S
197  69 OP
198  21  21
199  06  6

200  75  -
201  43 RCL
202  01  01
203  95  =
204  55  ÷
205  82 HIR
206  17  17
207  65  ×
208  53  (
209  22 INV
210  59 INT
211  69 OP
212  10  10
213  85  +
214  01  1
215  95  =
216  22 INV
217  44 SUM
218  01  01
219  61 GTO
220  01  01
221  94  94
222  76 LBL
223  11  A
224  37 P/R
225  09  9
226  69 OP
227  17  17
228  47 CMS
229  06  6
230  48 EXC
231  01  01
232  91 R/S
233  72 ST*
234  01  01
235  69 OP
236  21  21
237  61 GTO
238  02  02
239  32  32
```

Texas Instruments Inc.- You might have heard rumors or seen newspaper articles telling you about financial troubles at TI. It simply concerns an internal reorganization which implies a lay-off of about 6000 people. But, on the other hand, TI is actively advertizing jobs in the Dallas area for specialists in certain IC operations.

Recently, Forbes Magazine, in its March 15, 1982 issue had an article called "Texas Instruments at mid-life",by Anne Bagamery. Ms. Bagamery didn't have all of her facts straight, though. At one point she writes:"For example, TI is not increasing its commitments to the consumer products division -$ 19.95 watches and $ 17.50 calculators mainly - but is phasing the divsion out, with the exception of personal computers and its Speak and Spell learning aids, which embody a technological advance of tremendous promise."

Now, J. Fred Bucy, TI's president, just a week before announced that they were increasing their commitment to the consumer division. When shown the article in question by one of our members, who is also a large stockholder in TI, Mr.Bucy said: "We are continuing development on advanced programmable calculators and will soon have products to announce. I am sure you can appreciate my not being able to be more explicit until these developments are ready for general announcement. We continue to be the largest supplier of calculators to the U.S. market in both unit and dollar volume, and it is our intention to remain the leader in this area."

ZIMMERMAN'S PUZZLE: In v7n4p12 we presented this puzzle. Please refer to that issue for a complete explanation as to what was required.

The first solution I received when the newsletter was barely a few days in the mail came from Paul Blair in Milan, Illinois. His solution is short and sweet. It is also fast, although I didn't want to make this a speed contest. As is usual though, speed, compactness and elegance of solution are three factors that get you points.

000	76	LBL	005	53	(| 010 | 54 |)	015	55	÷	020	94	+/-	025	59	INT
001	11	A	006	24	CE	011	59	INT	016	05	5	021	85	+	026	65	×
002	85	+	007	55	÷	012	55	÷	017	95	=	022	01	1	027	01	1
003	05	5	008	01	1	013	02	2	018	22	INV	023	95	=	028	00	0
004	85	+	009	00	0	014	95	=	019	59	INT	024	22	INV	029	95	=
															030	91	R/S

Then the flood gates suddenly opened and I got heaps of solutions, mostly all good ones. Peter Brown in Denville, New Jersey sent me this one:

000	76	LBL	010	12	B	020	00	0	030	75	-	040	76	LBL	049	14	D
001	11	A	011	55	÷	021	95	=	031	01	1	041	13	C	050	92	RTN
002	29	CP	012	01	1	022	22	INV	032	00	0	042	85	+	051	76	LBL
003	42	STO	013	00	0	023	44	SUM	033	54	)	043	43	RCL	052	14	D
004	00	00	014	95	=	024	00	00	034	22	INV	044	01	01	053	85	+
005	22	INV	015	59	INT	025	53	(	035	77	GE	045	95	=	054	01	1
006	67	EQ	016	42	STO	026	02	2	036	13	C	046	94	+/-	055	00	0
007	12	B	017	01	01	027	65	×	037	75	-	047	22	INV	056	95	=
008	92	RTN	018	65	×	028	43	RCL	038	09	9	048	77	GE	057	92	RTN
009	76	LBL	019	01	1	029	00	00	039	95	=						

Then I received one from John Adams in Pittsburg, Pennsylvania. John says that he observed the pattern for the input numbers 00 through 09 and the fact that the single digit numbers decrease by 1 as you move from left to right anabled him to write the needed formulas to accomplish the task: When B=0 then the single digit= 0-A; when B=1,2,3,4 then the digit =10-2B-A; when B=5,6,7,8,9 then the digit = 19-2B-A; if the resulting gigit is negative then add 10 to the result.

000	76	LBL	010	00	0	020	95	=	029	85	+	038	12	B	047	29	CP
001	11	A	011	95	=	021	32	X:T	030	01	1	039	09	9	048	77	GE
002	59	INT	012	75	-	022	05	5	031	00	0	040	95	=	049	14	D
003	42	STO	013	59	INT	023	32	X:T	032	95	=	041	76	LBL	050	85	+
004	01	01	014	42	STO	024	67	EQ	033	29	CP	042	13	C	051	01	1
005	29	CP	015	02	02	025	12	B	034	77	GE	043	75	-	052	00	0
006	67	EQ	016	95	=	026	65	×	035	13	C	044	43	RCL	053	95	=
007	14	D	017	65	×	027	02	2	036	85	+	045	02	02	054	76	LBL
008	55	÷	018	01	1	028	94	+/-	037	76	LBL	046	95	=	055	14	D
009	01	1	019	00	0										056	92	RTN

The same day I received one from Clyde Durbin in Dallas, Texas. Clyde says he spent only one half hour on it, so if someone wants to refine it, that is OK. It works and that is that. Clyde also says that one could stack all the control digits in one register. I think unpacking would be rather slow, although the program could be made very efficient.

000	76	LBL	010	01	1	020	00	0	029	87	IFF	039	59	INT	049	29	CP
001	11	A	011	00	0	021	94	+/-	030	32	X:T	040	65	×	050	77	GE
002	42	STO	012	95	=	022	95	=	031	43	RCL	041	01	1	051	88	DMS
003	00	00	013	22	INV	023	77	GE	032	00	00	042	00	0	052	85	+
004	29	CP	014	59	INT	024	87	IFF	033	55	÷	043	95	=	053	01	1
005	67	EQ	015	75	-	025	85	+	034	01	1	044	59	INT	054	00	0
006	88	DMS	016	01	1	026	09	9	035	00	0	045	32	X:T	055	95	=
007	55	÷	017	95	=	027	95	=	036	95	=	046	75	-	056	76	LBL
008	06	6	018	65	×	028	76	LBL	037	85	+	047	32	X:T	057	88	DMS
009	32	X:T	019	01	1				038	22	INV	048	95	=	058	91	R/S

The reason I publish ALL these solutions is not so much that you should USE them, but rather that we all can learn something from the different approaches different programmers take. Programming is NOT a science governed by rigid rules, but rather an art in which every player takes part according to his/her own emotions.

Sid Hack in Columbia, South Carolina offers this one. Sid says that absolute addressing could speed this one up:

000	76	LBL	018	43	RCL	036	02	2	053	05	5	070	75	-	087	95	=
001	11	A	019	01	01	037	00	0	054	95	=	071	02	2	088	22	INV
002	47	CMS	020	75	-	038	95	=	055	42	STO	072	65	×	089	59	INT
003	42	STO	021	05	5	039	22	INV	056	04	04	073	53	(	090	65	×
004	01	01	022	95	=	040	59	INT	057	43	RCL	074	43	RCL	091	01	1
005	01	1	023	55	÷	041	65	×	058	03	03	075	02	02	092	00	0
006	00	0	024	02	2	042	02	2	059	22	INV	076	85	+	093	95	=
007	32	X:T	025	00	0	043	00	0	060	77	GE	077	43	RCL	094	91	R/S
008	43	RCL	026	95	=	044	95	=	061	38	SIN	078	04	04	095	76	LBL
009	01	01	027	59	INT	045	42	STO	062	08	8	079	95	=	096	30	TAN
010	65	×	028	42	STO	046	03	03	063	61	GTO	080	85	+	097	01	1
011	02	2	029	02	02	047	55	÷	064	39	COS	081	01	1	098	00	0
012	95	=	030	43	RCL	048	05	5	065	76	LBL	082	00	0	099	00	0
013	22	INV	031	01	01	049	95	=	066	38	SIN	083	95	=	100	44	SUM
014	77	GE	032	75	-	050	22	INV	067	09	9	084	55	÷	101	01	01
015	30	TAN	033	05	5	051	59	INT	068	76	LBL	085	01	1	102	61	GTO
016	76	LBL	034	95	=	052	65	×	069	39	COS	086	00	0	103	12	B
017	12	B	035	55	÷												

The next one comes from John Huntington Lewis in Norfolk, Virginia. Hunt says that the problem lies in recognizing that there are two orderly patterns in the digit generation table: a rotating sequence in the rows and another rotating -even,odd- in the columns. Observation of any row shows that the control digit decrements by one each time the tens digit of the key number increases. The algorithm then is "CONTROL DIGIT = BASE + OFFSET." Note the unique way Hunt has provided for the output: The input number is preserved and is printed together with the control digit as a decimal. If that is not wanted just omit LBL X^2 and write LBL X^2 R/S.

000	76	LBL	015	02	02	030	77	GE	045	09	9	060	54	)	075	29	CP
001	11	A	016	43	RCL	031	38	SIN	046	75	-	061	77	GE	076	76	LBL
002	58	FIX	017	01	01	032	65	×	047	53	(	062	30	TAN	077	33	X^2
003	01	01	018	22	INV	033	02	2	048	02	2	063	85	+	078	32	X:T
004	29	CP	019	59	INT	034	54	)	049	65	×	064	01	1	079	55	÷
005	42	STO	020	65	×	035	94	+/-	050	43	RCL	065	00	0	080	01	1
006	00	00	021	01	1	036	85	+	051	01	01	066	54	)	081	00	0
007	55	÷	022	00	0	037	01	1	052	54	)	067	76	LBL	082	54	)
008	01	1	023	54	)	038	00	0	053	54	)	068	30	TAN	083	85	+
009	00	0	024	42	STO	039	54	)	054	76	LBL	069	32	X:T	084	43	RCL
010	54	)	025	01	01	040	61	GTO	055	39	COS	070	01	1	085	00	00
011	42	STO	026	05	5	041	39	COS	056	29	CP	071	00	0	086	95	=
012	01	01	027	32	X:T	042	76	LBL	057	75	-	072	22	INV	087	99	PRT
013	59	INT	028	43	RCL	043	38	SIN	058	43	RCL	073	67	EQ	088	91	R/S
014	42	STO	029	01	01	044	01	1	059	02	02	074	33	X^2			

Then Lester Tibbetts from Emporium, Pennsylvania came through with the following one. He described the development of his algorithm almost in exactly the same words as Hunt Lewis did above, although he was able to do it in about half as many steps.

000	76	LBL	007	00	0	014	02	2	021	77	GE	028	43	RCL	035	01	1
001	11	A	008	75	-	015	00	0	022	34	FX	029	01	01	036	00	0
002	29	CP	009	59	INT	016	75	-	023	09	9	030	95	=	037	95	=
003	67	EQ	010	42	STO	017	01	1	024	94	+/-	031	94	+/-	038	76	LBL
004	33	X^2	011	01	01	018	00	0	025	85	+	032	77	GE	039	33	X^2
005	55	÷	012	95	=	019	85	+	026	76	LBL	033	33	X^2	040	91	R/S
006	01	1	013	65	×	020	22	INV	027	34	FX	034	85	+			

Robert Prins in Delft, the Netherlands, sent me three solutions, one shorter than the former one. His last one goes like this:

```
000  76 LBL    007  54  )    014  00  00    021  59 INT    028  53  (    035  85  +
001  11  A     008  42 STO    015  85  +     022  65  ×     029  43 RCL    036  77 GE
002  53  (     009  00  00    016  02  2     023  09  9     030  00  00    037  00  00
003  29 CP     010  53  (     017  49 PRD    024  75  -     031  75  -     038  40  40
004  55  ÷     011  59 INT    018  00  00    025  01  1     032  69 DP     039  01  1
005  01  1     012  94 +/-    019  43 RCL    026  00  0     033  10  10    040  00  0
006  00  0     013  44 SUM    020  00  00    027  65  ×     034  54  )    041  54  )
                                                                          042  92 RTN
```

His compatriot A.L. Koevoet sends this one. It requires you to load the number 975319753 into register 00, to enter the two-digit number and press A. The LBL E routine is optional if you want to list the entire table.

```
000  76 LBL    020  75  -    040  02  02    060  01  01    080  01  01    100  99 PRT
001  11  A     021  05  5    041  43 RCL    061  29 CP     081  99 PRT    101  92 RTN
002  42 STO    022  65  ×    042  00  00    062  43 RCL    082  92 RTN    102  76 LBL
003  01  01    023  43 RCL    043  65  ×    063  03  03    083  69 DP     103  15  E
004  99 PRT    024  02  02    044  43 RCL    064  67  EQ    084  21  21    104  00  0
005  55  ÷     025  54  )    045  02  02    065  00  00    085  01  1     105  42 STO
006  05  5     026  42 STO    046  22 INV    066  83  .     086  00  0     106  05  05
007  54  )     027  02  02    047  28 LOG    067  32 X:T    087  32 X:T    107  43 RCL
008  59 INT    028  43 RCL    048  52  EE    068  93  .     088  43 RCL    108  05  05
009  42 STO    029  01  01    049  22 INV    069  05  5     089  01  01    109  11  A
010  02  02    030  55  ÷    050  52  EE    070  77  GE    090  67  EQ    110  69 DP
011  55  ÷     031  02  2    051  54  )    071  00  00    091  00  00    111  25  25
012  04  4     032  00  0    052  22 INV    072  98  98    092  95  =     112  61 GTO
013  54  )     033  54  )    053  59 INT    073  01  1     093  99 PRT    113  01  01
014  22 INV    034  59 INT    054  65  ×    074  67  EQ    094  92 RTN    114  07  07
015  59 INT    035  44 SUM    055  01  1    075  00  00    095  00  0
016  42 STO    036  02  02    056  00  0    076  83  .     096  99 PRT
017  03  03    037  09  9    057  54  )    077  69 DP     097  92 RTN
018  43 RCL    038  22 INV    058  59 INT    078  31  31    098  43 RCL
019  01  01    039  44 SUM    059  42 STO    079  43 RCL    099  01  01
```

975319753. 00

Now George Thomson (no "p" please. The "p" indicates the wrong side of the tracks. George is Scottish, thus no "p.") has this very short one to offer:

```
000  76 LBL    006  82 HIR    012  00  0     018  22 INV    024  59 INT    030  59 INT
001  11  A     007  04  04    013  93  .     019  59 INT    025  55  ÷     031  65  ×
002  55  ÷     008  59 INT    014  09  9     020  65  ×     026  01  1     032  01  1
003  01  1     009  94 +/-    015  75  -     021  02  2     027  00  0     033  00  0
004  00  0     010  85  +     016  82 HIR    022  02  2     028  95  =     034  95  =
005  95  =     011  03  3     017  14  14    023  95  =     029  22 INV    035  92 RTN
```

And Charlie Williamson would solve it as follows:

```
000  76 LBL    014  42 STO    028  54  )    042  71 SBR    055  32 X:T    068  01  1
001  32 X:T    015  00  00    029  59 INT    043  32 X:T    056  54  )    069  00  0
002  32 X:T    016  55  ÷     030  53  (    044  54  )    057  53  (    070  54  )
003  65  ×     017  05  5     031  94 +/-    045  42 STO    058  24 CE    071  53  (
004  53  (     018  54  )     032  85  +     046  01  01    059  65  ×     072  32 X:T
005  32 X:T    019  53  (     033  01  1     047  53  (    060  02  2     073  01  1
006  55  ÷     020  59 INT    034  00  0     048  43 RCL    061  54  )    074  00  0
007  32 X:T    021  85  +     035  54  )    049  00  00    062  53  (    075  32 X:T
008  54  )     022  01  1     036  53  (    050  32 X:T    063  94 +/-    076  75  -
009  59 INT    023  54  )     037  32 X:T    051  05  5     064  85  +     077  71 SBR
010  92 RTN    024  53  (     038  01  1     052  32 X:T    065  43 RCL    078  32 X:T
011  76 LBL    025  24 CE    039  00  0     053  75  -     066  01  01    079  54  )
012  11  A     026  55  ÷     040  32 X:T    054  71 SBR    067  85  +     080  92 RTN
013  53  (     027  02  2     041  75  -
```

Nello Coda first sent this "conventional" solution. That is, one that conforms to the definition "computed" as stated in the original posing of the problem.

000	76	LBL	012	76	LBL	024	65	×	036	11	11
001	38	SIN	013	39	COS	025	01	1	037	59	INT
002	55	÷	014	85	+	026	00	0	038	94	+/-
003	01	1	015	43	RCL	027	95	=	039	44	SUM
004	00	0	016	11	11	028	91	R/S	040	11	11
005	95	=	017	95	=	029	76	LBL	041	55	÷
006	22	INV	018	55	÷	030	11	A	042	01	1
007	59	INT	019	01	1	031	55	÷	043	00	0
008	65	×	020	00	0	032	01	1	044	85	+
009	01	1	021	95	=	033	00	0	045	01	1
010	00	0	022	22	INV	034	95	=	046	95	=
011	95	=	023	59	INT	035	42	STO	047	22	INV

048	59	INT	060	02	2						
049	65	×	061	00	0						
050	01	1	062	85	+						
051	02	2	063	02	2						
052	32	X:T	064	00	0						
053	01	1	065	95	=						
054	00	0	066	77	GE						
055	95	=	067	38	SIN						
056	48	EXC	068	75	-						
057	11	11	069	01	1						
058	94	+/-	070	61	GTO						
059	65	×	071	39	COS						

But, as Nello points out, this one is only a special case, which is "computable." Suppose your control digits were arranged in such a random order that it would be virtually impossible to compute a table.
Now, before I hear a chorus of howls go up accompanied by cries of "foul" and "this does not fit the definition" let me entertain you with a short story:
My job entails "problem solving". I have been doing this for the last thirty years and I am slowly getting the hang of it. Now, the medical researchers I work for and with are supposed to define the problems for me so I can work them out. Invariably I get either a problem with severe restrictions, special cases or partial solutions. When I was still young and inexperienced I would fall for these ploys and give the researchers exactly what they wanted, a special case solution. After about one month the instrument I built for them would be returned with the plea "can you modify it for the following conditions." If I did, it would return after another two months, for more modifications. I once drew a picture of my own face, thick drops of bloody sweat dripping from it, with the caption NO MORE MODIFICATIONS, PLEASE. Thus, I learned that it is much better to give a GENERAL solution, EVEN IF ONLY A PARTIAL SOLUTION WAS REQUESTED.
User instructions for Nello's general solution: Put the calculator in 10 OP 17 and load the ten registers with the control digits as shown. Key in the program and record everything in 6 OP 17 on side 1 of a mag card. Enter the two-digit number and press A.
You see now that by loading the control digits in a different order in the regs, any random sequence can be accomodated. It will also be rather simple to relocate this routine and even use different registers. You might have to change a few constants in the routine itself, such as the 90 in steps 006-007.

135792468.	90
24681357.9	91
913570246.8	92
802469135.7	93
791358024.6	94
680247913.5	95
579136802.4	96
468025791.3	97
357914680.2	98
246803579.1	99

000	76	LBL	010	11	11	020	95	=	030	65	×
001	11	A	011	22	INV	021	22	INV	031	06	6
002	55	÷	012	59	INT	022	28	LOG	032	69	OP
003	01	1	013	65	×	023	35	1/X	033	17	17
004	00	0	014	01	1	024	65	×	034	01	1
005	85	+	015	00	0	025	73	RC*	035	00	0
006	09	9	016	69	OP	026	11	11	036	95	=
007	00	0	017	17	17	027	95	=	037	59	INT
008	95	=	018	01	1	028	22	INV	038	91	R/S
009	42	STO	019	00	0	029	59	INT			

And this is where I stop for this issue. Not that I ran out of submissions: at the time of this writing, May 25th, I have received an additional 18 solutions. This is by far the most successful "utmaning" we ever ran in the TI PPC NOTES. If some of the other solutions prove to be really different from the ones published in this issue, I will put them in next issue. Otherwise, I will consider this poney beaten to death. To the many contributors: thanks. Please find Harald M. Otto's new puzzle somewhere in this issue. You will know what to do with. The best of luck.
