EASTER MASTER

EASTER MASTER

001	*LBL A	21 11	076	GTO 1	22 01	151	-	-45
002	CF 3	16 22 03	077	X=Y?	16-33	152	RCL C	36 13
003	CLR G	16-53	078	GTO 3	22 03	153	x	-35
004	STO 1	35 46	079	R↓	-31	154	RND	16 24
005	DSP 0	-63 00	080	GTO 1	22 01	155	X=0?	16-43
006	F1?	16 23 01	081	*LBL 3	21 03	156	RCL C	36 13
007	PRTX	-14	082	RCL 5	36 05	157	RTN	24
008	1	01	083	+	-55	158	*LBL 0	21 00
009	%	55	084	*LBL 1	21 01	159	1	01
010	INT	16 34	085	STO E	35 15	160	STO 5	35 05
011	STO 8	35 08	086	2	02	161	R↓	-31
012	1	01	087	3	03	162	RTN	24
013	6	06	088	X≠Y	-41	163	*LBL 4	21 04
014	-	-45	089	X≤Y?	16-35	164	0	00
015	X<0?	16-45	090	SF 0	16 21 00	165	STO C	35 13
016	CLX	-51	091	7	07	166	R↓	-31
017	STO 2	35 02	092	5	05	167	RTN	24
018	4	04	093	X≠Y	-41	168	*LBL B	21 12
019	+	-24	094	-	-45	169	SF 3	16 21 03
020	INT	16 34	095	F0?	16 23 00	170	STO 1	35 46
021	STO A	35 11	096	GSB 5	23 05	171	F1?	16 23 01
022	1	01	097	STO B	35 12	172	PRTX	-14
023	+	-55	098	5	05	173	4	04
024	+	-55	099	7	07	174	+	-24
025	RCL I	36 46	100	RCL E	36 15	175	INT	16 34
026	4	04	101	-	-45	176	RCL I	36 46
027	+	-24	102	F0?	16 23 00	177	+	-55
028	INT	16 34	103	GSB 5	23 05	178	4	04
029	+	-55	104	CF 0	16 22 00	179	+	-55
030	RCL 2	36 02	105	7	07	180	GSB 9	23 09
031	-	-45	106	STO C	35 13	181	GSB 2	23 02
032	GSB 9	23 09	107	GSB 6	23 06	182	3	03
033	GSB 2	23 02	108	RCL 3	36 03	183	-	-45
034	1	01	109	X≠Y	-41	184	3	03
035	0	00	110	X≤Y?	16-35	185	0	00
036	-	-45	111	GSB 4	23 04	186	STO C	35 13
037	3	03	112	-	-45	187	GSB 6	23 06
038	0	00	113	RCL B	36 12	188	STO E	35 15
039	STO C	35 13	114	+	-55	189	GTO 1	22 01
040	+	-24	115	RCL C	36 13	190	*LBL 2	21 02
041	FRC	16 44	116	+	-55	191	RCL I	36 46
042	RCL C	36 13	117	DSP 2	-63 02	192	1	01
043	x	-35	118	9	09	193	+	-55
044	STO 7	35 07	119	3	03	194	1	01
045	RCL 8	36 08	120	+	-55	195	9	09
046	1	01	121	3	03	196	STO E	35 13
047	7	07	122	1	01	197	GSB 6	23 06
048	-	-45	123	.	-62	198	1	01
049	2	02	124	1	01	199	1	01
050	5	05	125	STO C	35 13	200	X≠Y	-41
051	+	-24	126	GSB 6	23 06	201	X>Y?	16-34
052	INT	16 34	127	1	01	202	GSB 0	23 00
053	ST-8	35-45 08	128	%	55	203	x	-35
054	1	01	129	RCL 4	36 04	204	RTN	24
055	5	05	130	+	-55	205	*LBL 9	21 09
056	ST-8	35-45 08	131	F1?	16 23 01	206	7	07
057	3	03	132	GTO 7	22 07	207	+	-24
058	ST+8	35-24 08	133	RTN	24	208	FRC	16 44
059	RCL 8	36 08	134	*LBL 7	21 07	209	7	07
060	INT	16 34	135	PRTX	-14	210	x	-35
061	RCL 7	36 07	136	SPC	16-11	211	RND	16 24
062	+	-55	137	ISZ I	16 26 46	212	7	07
063	RCL A	36 11	138	RCL I	36 46	213	X≠Y	-41
064	+	-55	139	F3?	16 23 03	214	-	-45
065	RCL 2	36 02	140	GTO B	22 12	215	STO 3	35 03
066	-	-45	141	GTO A	22 11	216	RTN	24
067	RCL C	36 13	142	*LBL 5	21 05	217	*LBL C	21 13
068	GSB 6	23 06	143	RCL C	36 13	218	SF 1	16 21 01
069	2	02	144	-	-45	219	1	01
070	4	04	145	RTN	24	220	RTN	24
071	X=Y?	16-33	146	*LBL E	21 06	221	*LBL D	21 15
072	SF 2	16 21 02	147	+	-24	222	CF 1	16 22 01
073	1	01	148	ENT↑	-21	223	0	00
074	+	-55	149	INT	16 34	224	RTN	24
075	F2?	16 23 02	150	STO 4	35 04			

In 1904 Easter fell on April 3, as you can see by loading the program elsewhere in this issue, keying "1904" and pressing "A"; after about 13 seconds you will see "4.03". Also Easter 1979 will fall on April 15 and Easter 2001 on April 15 also. In 3695, if the present system is still in effect, they will celebrate Easter on April 17.

"The present system" means the Gregorian Calendar, which has governed British and American Easters since 1753, though France, Italy, Spain and much of Germany had adopted it in time for Easter 1583. The Explanatory Supplement to the *American Ephemeris* gives a table (pp. 414-416) of the dates on which the Gregorian Calendar was adopted in various parts of the world (1918 in Russia!). Be-

fore that they used the Julian Calendar, which the program also incorporates; just press "B". Easter 1253 fell on April 20.

To print out a table of Easter Dates use the toggle "C", which shows 0 for single dates, 1 for table printout. Press "C" to obtain the "1" signal, key the starting date, then "A" if Gregorian and "B" if Julian.

The program has been checked for accuracy over a span of 171 Gregorian years (1850 - 2020) and 82 Julian years (1200 - 1281). One needs to check long spans because special conditions can arise at rather long intervals. The algorithms are too complicated to explain here. You can reconstruct them by flowcharting the program, or you can look them up in a reprint of *A Budget of Paradoxes* by the British mathematician Augustus De Morgan, who devised them about 140 years ago.

Briefly, the program works by computing the Epact for the year (see Dan Fenstermacher's explanation and program, V5 N1 P16) by a method which is longer than Dan's but isn't restricted to a span of 418 years, and yields as byproducts the Golden Number and a numerical equivalent of the Dominical Letter. Out of these three quantities it then concocts the date. The Epact is in RE at the end of the run.

The Julian Epact is rather simple (steps 168 - 189). The Gregorian Epact (steps 1 - 83 plus several subroutines) has many more special cases to correct. At various points we need numbers modulo 7, 19, 30, or 31; LBL 6 is a "mod" subroutine that works on a modulus stored in RC. At step 116 we have obtained the number of days till Easter, counting from March 1; the rest of the run, down to step 133, converts this into a date in the format M.DD. The reason this uses 31.1 instead of 31 as a modulus is to prevent March 31 being misrepresented as April 31. The flags control printing (F1), trigger a correction routine (F2), and tell the table loop whether the program is in the Gregorian or the Julian mode (F3). The registers are cleared (step 3) because on each pass of the Gregorian loop R5 must be zero for the rare occasions on which it gets added in (steps 77-8 and LBL 3); when it's added it must, according to certain conditions, be either zero or one (steps 201-2 and LBL 0). This has no significance for the Julian loop, which accordingly does without a CLR REG. Don't modify anything unless you're very sure it won't generate a wrong answer 50 years away from the years you spot-check.

—Hugh Kenner (103)