CALENDAR ALGORITHMS

There are several well known algorithms used for determining the day of the week for any date. The calendar programs included here are based on the following:

$$DOW = \left| \text{INT}\left(\frac{XY}{12}\right) + R + \text{INT}\left(\frac{R}{4}\right) + M + D + \left(6 - 2\left(\left|\sqrt{AB}\right|\right)\right) \right|_7$$

where: DOW = day of week
INT = the integral value of
R = the remainder after the indicated division
D = the day of the month
XY = the last two digits of the year
AB = the first two digits of the year
7 = modulus of the enclosed modular sum
(means the same as "cast out sevens")
4 = denotes the modular value of the enclosed AB
with 4 the modulus. (cast out fours)
M = a month number from the table below

JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
1	4	4	0	2	5	0	3	6	1	4	6

The algorithm above applies only to the Gregorian calendar which began on Friday, October 15, 1582. The ten days immediately previous were omitted from the calendar, the day just before being Thursday, October 4, 1582 in the Julian calendar.

Do not confuse the Julian calendar with Julian Day numbers which are used in H-P's CALENDAR FUNCTIONS program (SD-04A). There is little or no relation between the two, in spite of the similarity of the names. The Julian calendar was established in the Roman empire by Julius Caesar with the advice of the astronomer Sosigenes to correct the confusion that had arisen between the year and the seasons. The year 46 B.C. was increased by intercalations to 445 days and his calendar was begun on 1-1-45 B.C. Due to the confusion resulting from it's adoption the intercalary days were incorrectly added until the error was corrected by Augustus Caesar in 8 B.C. who omitted further intercalations until A.D. 8. After 1-1-8 the calendar remained in this form until the Gregorian reform in 1582.

The Julian calendar assumed the year tobe 365.250 days and corrected the discrepancy by adding an extra day every fourth year. Since the exact value of the fractional part is .24219878 an error slowly accumulated until the reform in 1582. That last reform consisted of:

1. Dropping 10 days from the calendar
(correcting the accumulated error)
2. Omitting the intercalary day in those century years not evenly divisible by 400.
Thus 1700, 1800, 1900, 2100 etc. are not
leap years but 1600, 2000 are.

In the sixteenth century Josephus Justus Scaliger in honour of his father Julian , introduced the continuous era known as the Julian period. It is a continuous succession of days--no years, no months. It began 1-1-4713 B.C. The Julian day begins at noon. Since all days are consecutively numbered, the obvious value to astronomers and others is to determine the elapsed days between two dates. Merely subtract the corresponding Julian day numbers. Tables and calculator programs are available to determine the Julian day number for any date in history or the future.

The program entitled Julian calendar computes the day of the week for any date in the period 1-1-8 to and including 10-4-1582. For reasons stated above it may be erroneous back to 1-1-1. Be sure to enter a date such as 1-1-8 in this manner: 1.010008

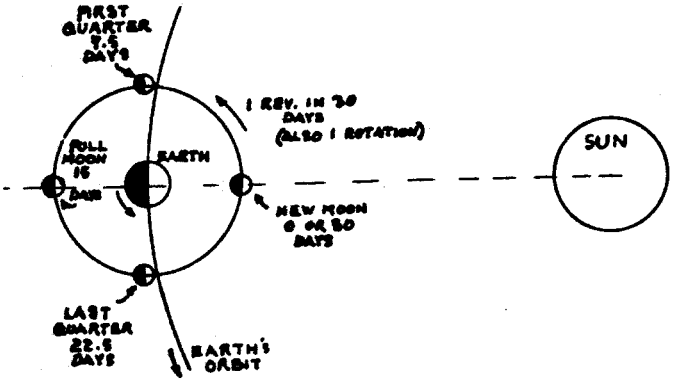
FRIDAY 13th Rearranges the above algorithm to search the calendar for consecutive occurrences of a given week day on a given day of the month such as Fri 13th, Mon.1st etc.

CALENDAR SEARCH examines consecutive years for the occurrence of a given date on a given week day, such as Christmas on Sunday or July 4th on a Saturday. Expect to be surprised with Feb.29th on Sunday.

MOON PHASE-- For those not familiar with the jargon the little sketch below should serve to explain the "age" or phase of the moon :

FURTHER READING

Sky and Sextant. Practical Celestial Navitation, by John P. Budlong, is a 1975 copyrighted book with a special chapter on the use of hand-held calculators. The SR-50, HP-35, HP-45. Van Nostrand Reinhold Co., 5-3/4 X 8-1/2", hard-bound, 151 pages.



The phase of the moon for any date can be determined with the following simple algorithm:

$$AGE = M + D + E \quad (\text{mod } 30)$$

(denotes modular sum, modulus 30)

where: D = the day of the month
E = the EPACT for the year (see below)
M = the month number from the following table:

JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	2	0	2	2	4	4	6	7	8	9	10

Do not confuse this series of numbers with those used in calendar algorithms. Note that the last four months were named for their numbers.

The EPACT is the age of the moon less one day on Jan.1st in the ecclesiastical lunar calendar. It can have any value from 0 to 29. It usually, but not always increases by 11 each year, a modular sum with 30 the modulus, (when the increase exceeds thirty, discard the thirty and use the remainder. The epact for 1977 is 10, 1978 is 21, and 1979 is 2, etc. The program Moon Phase, by laborious corrections derives the proper value of the Epact for each year over the interval 1582--2000.

Since the Epact is used to determine the dates of Easter and other religious calenders, the program entitled EPACT derives serial values over the same interval.

KAPREKAR'S CONSTANT-- (See Martin Gardner's articles in Scientific American) Kaprekar discovered the unique, but like perfect numbers useless fact that if any 4 digit number (provided that at least one digit be different from the rest) is arranged in ascending order and subtracted from the same digits arranged in descending order, a new series of digits is formed and upon repeating the process upon the remainder -- in eight or less iterations the remainder becomes the constant 6174. --One readily finds that there is also a three digit constant, but I have yet to discover one for five or more. Perhaps some member with a flair for number theory could prove no others exist?

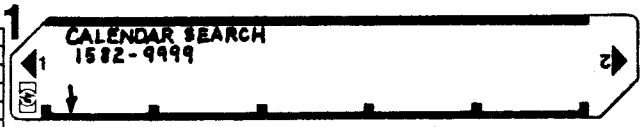
See programs starting on the next page. Dan M. Fenstermacher (1438)

STEP	KEY ENTRY 1	KEY CODE
1	LBL A	
	RCL I	
	4	
	+	
5	FRC	
	X=0?	
	GTO 2	
	GTO 3	
	LBL E	
10	SF 2	
	RCL I	
	4	
	0	
	0	
15	+	
	FRC	
	X=0?	
	GTO 4	
	GTO 2	
20	LBL 8	
	RCL I	
	EEX	
	2	
	+	
25	FRC	
	X=0?	
	GTO 4	
	GTO 3	
	LBL 4	
30	CF 2	
	GTO 9	
	LBL 3	
	RCL I	
	EEX	
35	2	
	÷	
	STO 9	
	INT	
	4	
40	+	
	FRC	
	8	
	X	
	6	
45	X≠Y	
	−	
	STO A	
	RCL 9	
49	FRC	
50	EEX	
	2	
	X	
	1	
	2	
55	+	
	STO 8	
	INT	
	STO 7	
	RCL 8	
60	FRC	
	1	
	2	
	X	
	STO+7	
65	4	
	+	
	RND	
	INT	
	STO+7	
70	RCL 2	
	STO 8	
	RCL A	
	STO+8	
	RCL 1	
75	1	
	X=Y?	
	GTO B	
	RCL 1	
	2	
80	X=Y?	
	GTO 0	
	GTO 5	
	LBL 0	
	RCL 2	
85	2	
	9	
	X=Y?	
	GTO 9	
	GTO C	
90	LBL 9	
	FS?	
	GTO 9	
	GTO 1	
	LBL 2	
95	GSB 7	
	GTO C	
	LBL 5	
	RCL 1	
	3	
100	X=Y?	

STEP	KEY ENTRY 2	KEY CODE
101	GTO C	
	RCL 1	
	4	
	X=Y?	
105	GTO D	
	RCL 1	
	5	
	X=Y?	
	GTO E	
110	RCL 1	
	6	
	X=Y?	
	GTO 6	
	RCL 1	
115	7	
	X=Y?	
	GTO D	
	RCL 1	
	8	
120	X=Y?	
	GTO 6	
	RCL 1	
	9	
	X=Y?	
125	GTO C	
	RCL 1	
	1	
	0	
	X=Y?	
130	GTO B	
	RCL 1	
	1	
	1	
	X=Y?	
135	GTO C	
	GTO C	
	LBL B	
	1	
	STO+8	
140	FS?	
	GSB 7	
	GTO d	
	LBL C	
	4	
145	STO+8	
	FS?	
	GSB 7	
	GTO d	
149	LBL D	
150	0	
	STO+8	
	GTO d	
	LBL E	
	2	
155	STO+8	
	GTO d	
	LBL A	
	5	
	STO+8	
160	GTO d	
	LBL b	
	3	
	STO+8	
	GTO d	
165	LBL c	
	6	
	STO+8	
	GTO d	
	LBL d	
170	RCL 8	
	7	
	+	
	FRC	
	7	
175	X	
	STO 4	
	RCL 7	
	7	
	+	
180	FRC	
	7	
	X	
	RND	
	STO 5	
185	RCL 4	
	+	
	7	
	+	
	FRC	
190	7	
	X	
	RND	
	STO 6	
	RCL 3	
195	X=Y?	
	GTO A	
	GTO 1	
	LBL c	
	RCL 2	
200	EEX	

STEP	KEY ENTRY 1
201	2
	+
	RCL 1
	+
205	RCL I
	EEX
	4
	+
	+
210	PRT X
	GTO 1
	LBL 1
	192 I
	GTO A
215	LBL 7
	1
	STO-8
	RTN
	R/S
220	
224	

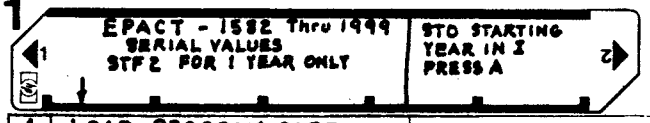
1. CALENDAR SEARCH BY: DAN M. FENSTERMACHER - 1438



97

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	LOAD PROGRAM CARD			
	SIDES 1 & 2			
2	STORE STARTING YEAR IN			
	I REGISTER.			
	STORE MONTH IN REG. 1			
	JAN.=1, FEB.=2 ETC.			
	STORE DAY OF MONTH IN			
	REGISTER 2			
	STORE DAY OF WEEK IN			
	REG. 3. SUN=1, -SAT=0			
3	PRESS A - OUTPUT IS SERIAL			MM.DDYYYY
	LIST OF YEARS HAVING THE			
	GIVEN DATE ON THE GIVEN			
	WEEK DAY.			
	EXAMPLE: FEB. 29 OCCURS ON SUNDAY ONLY			THREE
	TIMES IN THIS CENTURY.			

1. EPACT BY: DAN M. FENSTERMACHER - 1438

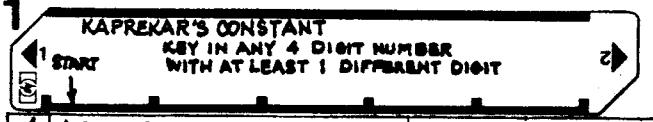


97

1	LOAD PROGRAM CARD			
	SIDE 1 ONLY.			
2	STORE STARTING YEAR			
	IN REGISTER I. SET FLAG			
	2 FOR 1 YEAR ONLY.			
	PROGRAM PRINTS ERROR			
	FOR YEARS PAST 1999, &			
	IS INVALID FOR JULIAN ERA.			
3	PRESS A.			YEAR
	OUTPUT IS YEARS IN			EPACT
	SERIAL ORDER FOLLOWED			&PC.
	BY EPACT FOR EACH YEAR			
	THE EPACT IS DEFINED AS THE AGE OF THE MOON			
	LESS 1 DAY ON JAN. 1 OF EACH YEAR IN THE			
	ECCLESIACTICAL LUNAR CALENDAR. SEE TEXT.			

SEE PAGE 22 FOR PROGRAM LISTING (RIGHT SIDE)

1. KAPREKAR'S CONSTANT BY: DAN M. FENSTERMACHER - 1438



97

1	LOAD PROGRAM CARD -			
	SIDES 1 & 2			
2	KEY IN ANY 4 DIGIT			
	NUMBER WITH AT LEAST			
	ONE DIGIT DIFFERENT FROM			
	THE REST. EXAMPLES:			
	0001, 9366, 8881 ETC.			
3	PRESS A. OUTPUT IS INPUT			
	NUMBER ARRANGED IN DES-			
	CENDING ORDER, THEN IN			
	ASCENDING ORDER - THE TWO			
	SUBTRACTED AND THE PRO-			
	CEDURE REPEATED UPON THE			
	REMAINDER FOR UP TO 9			
	ITERATIONS UNTIL A REPEATING			
	CONSTANT IS OBTAINED.			
	PROGRAM HALTS WHEN CONSTANT 6174 IS OBTAINED.			

SEE PAGE 22 FOR PROGRAM LISTING (LEFT SIDE)

STEP	KEY ENTRY 1	KEY CODE
1	LBL A	
	DSP 0	
	PRT X	
	STO 1	
5	EEX	
	9	
	+	
	STO E	
	INT	
10	STO A	
	RCL E	
	FRC	
	EEX	
	1	
15	X	
	STO 3	
	INT	
	STO B	
	RCL 3	
20	FRC	
	EEX	
	1	
	X	
	STO 4	
25	INT	
	STO C	
	RCL 4	
	FRC	
	EEX	
30	1	
	X	
	INT	
	STO D	
	GTO B	
35	LBL B	
	RCL A	
	RCL B	
	X≤Y?	
	X≠Y	
40	RCL C	
	X≤Y?	
	X≠Y	
	RCL D	
	X≤Y?	
45	X≠Y	
	STO 9	
	R↓	
	STO 1	
49	R↓	
50	STO 2	
	R↓	
	STO 3	
	CLX	
	RCL 3	
55	RCL 2	
	X≤Y?	
	X≠Y	
	RCL 1	
	X≤Y?	
60	X≠Y	
	STO 8	
	CLX	
	R↓	
	X≤Y?	
65	X≠Y	
	STO 7	
	R↓	
	STO 6	
	RCL 9	
70	EEX	
	3	
	X	
	Σ+	
	RCL 8	
75	EEX	
	2	
	X	
	Σ+	
	RCL 7	
80	EEX	
	1	
	X	
	Σ+	
	RCL 6	
85	Σ+	
	RCL E	
	STO E	
	Σ-	
	4	
90	1	
	7	
	4	
	STO 4	
	RCL 6	
95	EEX	
	3	
	X	
	STO 5	
	RCL 7	
100	EEX	

STEP	KEY ENTRY 2	KEY CODE
101	2	
	X	
	STO + 5	
	RCL 9	
105	EEX	
	1	
	X	
	STO + 5	
	RCL 9	
110	STO + 5	
	RCL E	
	PRT X	
	RCL 5	
	PRT X	
115	−	
	PRT X	
	RCL 4	
	X=Y?	
	R/S	
120	X≠Y	
	SPC	
	GTO A	
	R/S	
125		
130		
135		
140		
145		
149		
150		
55		
60		
65		
70		
75		
80	KAPREKAR'S	
	CONSTANT	
85	SEE PAGE 21	
	FOR INSTRUCTIONS	
90		
95		
100		

STEP	KEY ENTRY 1	KEY CODE
1	LBL A	
	1	
	2	
	3	
5	STO 9	
	2	
	2	
	3	
	STO 8	
10	3	
	2	
	3	
	STO 7	
	4	
15	2	
	3	
	STO 6	
	1	
	5	
20	7	
	7	
	STO 5	
	LBL 4	
	RCL I	
25	PRT X	
	RCL 2	
	−	
	STO 3	
	1	
30	1	
	X	
	9	
	0	
	+	
35	FRC	
	3	
	0	
	X	
	STO 4	
40	1	
	STO + 4	
	GTO B	
	LBL b	
	RCL 3	
45	1	
	9	
	+	
	INT	
49	STO + 4	
50	GTO E	
	LBL E	
	RCL 4	
	3	
	0	
55	+	
	FRC	
	3	
	0	
	X	
60	DSP 0	
	RND	
	PRT X	
	SPC	
	FS? 1	
65	R/S	
	ISZ I	
	GTO 4	
	LBL B	
	RCL 3	
70	RCL 9	
	X>Y?	
	GTO b	
	GTO C	
	LBL C	
75	RCL 3	
	RCL 8	
	X>Y?	
	GTO 1	
	GTO D	
80	LBL D	
	RCL 3	
	RCL 7	
	X>Y?	
	GTO 1	
85	GTO 2	
	LBL 2	
	RCL 3	
	RCL 6	
	X>Y?	
90	GTO 0	
	GTO 3	
	LBL 3	
	0	
	ENT↑	
95	+	
	R/S	
	LBL 1	
	1	
	STO − 4	
100	GTO b	

STEP	KEY ENTRY 2	KEY CODE
101	LBL 0	
	2	
	STO − 4	
	GTO b	
105	R/S	
110		
115		
120		
125		
130		
135		
140		
145		
149		
150		
55		
60		
65		
70		
75		
80	EPAC	
	SEE PAGE 21	
85	FOR INSTRUCTIONS	
90		
95		
100		