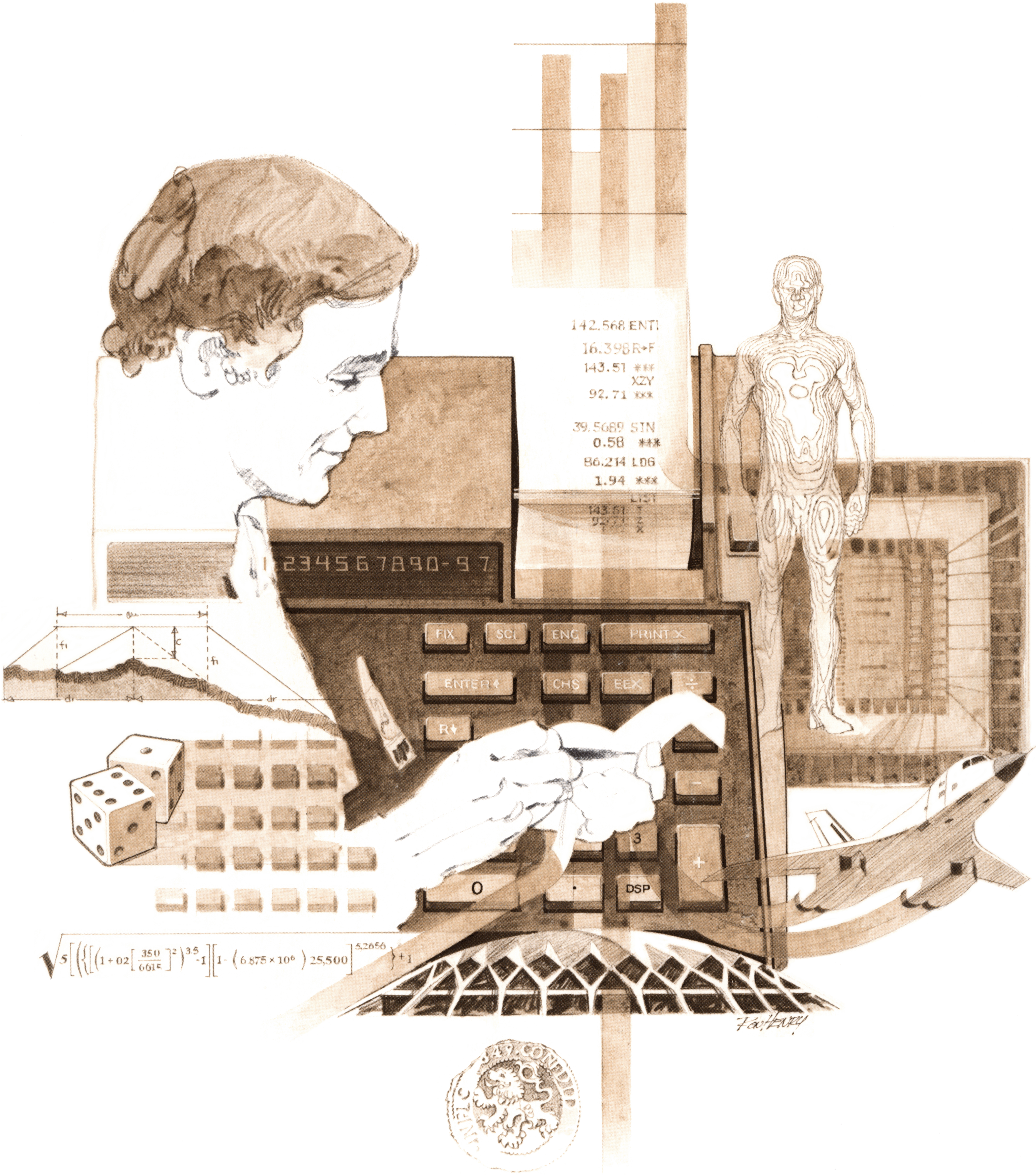
HEWLETT-PACKARD

HP-67/HP-97

Users' Library Solutions

Calendars



INTRODUCTION

In an effort to provide continued value to it's customers, Hewlett-Packard is introducing a unique service for the HP fully programmable calculator user. This service is designed to save you time and programming effort. As users are aware, Programmable Calculators are capable of delivering tremendous problem solving potential in terms of power and flexibility, but the real genie in the bottle is program solutions. HP's introduction of the first handheld programmable calculator in 1974 immediately led to a request for program **solutions** — hence the beginning of the HP-65 Users' Library. In order to save HP calculator customers time, users wrote their own programs and sent them to the Library for the benefit of other program users. In a short period of time over 5,000 programs were accepted and made available. This overwhelming response indicated the value of the program library and a Users' Library was then established for the HP-67/97 users.

To extend the value of the Users' Library, Hewlett-Packard is introducing a unique service—a service designed to save you time and money. The Users' Library has collected the best programs in the most popular categories from the HP-67/97 and HP-65 Libraries. These programs have been packaged into a series of low-cost books, resulting in substantial savings for our valued HP-67/97 users.

We feel this new software service will extend the capabilities of our programmable calculators and provide a great benefit to our HP-67/97 users.

A WORD ABOUT PROGRAM USAGE

Each program contained herein is reproduced on the standard forms used by the Users' Library. Magnetic cards are not included. The Program Description I page gives a basic description of the program. The Program Description II page provides a sample problem and the keystrokes used to solve it. The User Instructions page contains a description of the keystrokes used to solve problems in general and the options which are available to the user. The Program Listing I and Program Listing II pages list the program steps necessary to operate the calculator. The comments, listed next to the steps, describe the reason for a step or group of steps. Other pertinent information about data register contents, uses of labels and flags and the initial calculator status mode is also found on these pages. Following the directions in your HP-67 or HP-97 **Owners' Handbook and Programming Guide,** "Loading a Program" (page 134, HP-67; page 119, HP-97), key in the program from the Program Listing I and Program Listing II pages. A number at the top of the Program Listing indicates on which calculator the program was written (HP-67 or HP-97). If the calculator indicated differs from the calculator you will be using, consult Appendix E of your **Owner's Handbook** for the corresponding keycodes and keystrokes converting HP-67 to HP-97 keycodes and vice versa. No program conversion is necessary. The HP-67 and HP-97 are totally compatible, but some differences do occur in the keycodes used to represent some of the functions.

A program loaded into the HP-67 or HP-97 is not permanent—once the calculator is turned off, the program will not be retained. You can, however, permanently save any program by recording it on a blank magnetic card, several of which were provided in the Standard Pac that was shipped with your calculator. Consult your **Owner's Handbook** for full instructions. A few points to remember:

> The Set Status section indicates the status of flags, angular mode, and display setting. After keying in your program, review the status section and set the conditions as indicated before using or permanently recording the program.
>
> REMEMBER! To save the program permanently, **clip** the corners of the magnetic card once you have recorded the program. This simple step will protect the magnetic card and keep the program from being inadvertently erased.

As a part of HP's continuing effort to provide value to our customers, we hope you will enjoy our newest concept.

TABLE OF CONTENTS

Program Description I

Program Title Calendar Date to Julian Date Conversion
-(See the companion routine: Julian Date to Calendar Conversion)-

Contributor's Name Rayner K. Rosich
Address 7031 Pierson Street
City Arvada **State** Colorado **Zip Code** 80004

Program Description, Equations, Variables This algorithm converts any given calendar date (YYYY.MMDD) to a Julian date--a continuous count of days from an epoch in the very distant past. For example, Jan. 1, 1970, (1970.0101) yields a Julian date 2440588. Thus the interval between any two calendar dates (on the Gregorian Calendar) can be found by obtaining the Julian date for each and then finding the difference.

The algorithm implements the FORTRAN arithmetic statement function:
(I = year; J = month, 1-12: K = day of month)

$$\left\{\begin{array}{l} JD\ (I,\ J,\ K) = K - 32075 + 1461 * (I + 4800 + (J-14)/12)/4 \\ \quad + 367 * (J - 2 - ((J - 14)/12) * 12)/12 \\ \quad - 3 * ((I + 4900 + (J-14)/12)/100)/4, \end{array}\right\}$$

where the algorithm takes pains to implement the truncation feature of integer arithmetic in FORTRAN by use of the "f INT" operation on the HP-65.

The algorithm given here is a conversion * to the HP-65 of the algorithm of Fliegel and Van Flandern given in the Reference(s) Section below.

Operating Limits and Warnings The algorithm is valid for any calendar date (Gregorian Calendar Date) which produces a Julian Date greater than zero. See the reference for further details.

* This is submitted with the approval of the second author (TCV-F) of the ref.

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.

Program Description I

Program Title Julian Date to Calendar Date Conversion
-(See the companion routine: Calendar Date to Julian Date Conversion)-

Contributor's Name Rayner K. Rosich
Address 7031 Pierson Street
City Arvada **State** Colorado **Zip Code** 80004

Program Description, Equations, Variables This algorithm converts any given Julian date-a continuous count of days from an epoch in the very distant past--to a calendar date of the form YYYY.MMDD. For example, 2440588 yields 1970.0101 (Jan. 1, 1970). Thus the calendar date corresponding to an elapsed number of days from a given calendar date can be found by obtaining the final Julian date (by addition of the no. days elapsed to the Julian date corresponding to the given calendar date--see companion routine: Calendar Date to Julian Date Conversion) and then conversion to the final calendar date.

The algorithm implements the following FORTRAN subroutine (JD = Julian date input; Output: I=year, J=month [1-12]. K=day number in month):

```
Subroutine Date(JK,I,J,K)
L = JD + 68569
N = 4*L/146097
L = L - (146097 * N + 3)/4
I = 4000 * (L+1)/1461001
L = L - 1461 * I/4 +31
J = 80 * L/2447
K = L - 2447 * J/80
L = J/11
J = J + 2 = 12*L
I = 100* (N-49) + I + L
RETURN
END
```

The algorithm takes pains to implement the truncation feature of integer arithmetic in FORTRAN by use of the "f INT" operation on the HP-65.

The algorithm given here is a conversion* to the HP-65 of the algorithm of Fliegel and Van Flandern given in the Reference(s) section below.

Operating Limits and Warnings The algorithm is valid for any Julian date which is greater than zero. See the reference for further details.

* This is submitted with the approval of the second author (TCV-F) of the ref.

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.

Program Description II

Calendar Date to Julian Date Conversion

Sketch(es)

Sample Problem(s)

1. January 1, 1970 --- Julian date 2,440,588.
2. Determine the number of elapsed days between Aug. 28, 1940 and Aug. 28, 1975. The answer is: 12,783 days.
3. Julian date 2,440,588 --- 1970.0101 (Jan. 1, 1970)
4. Determine the date corresponding to 12,783 days elapsed from Aug. 28, 1940 (Julian date: 2429870.) The answer is: Aug. 28, 1975.

Note: The routine automatically sets the display for the format yyyy.MMDD of the result.

Solution(s)

Problem 1:

1970.0101 A → 2440588.00

Problem 2:

1970.0828 A → 2429870.00

$\boxed{\text{STO}}$ $\boxed{9}$ → 2429870.00

1975.0825 A → 2442653.00

$\boxed{\text{RCL}}$ $\boxed{9}$ $\boxed{-}$ → 12783.00

Problem 3:

1) Load Card 1.
2) 2440588 E → 1970.0101 Jan.1, 1970 (yyyy.MMDD)

Problem 4:

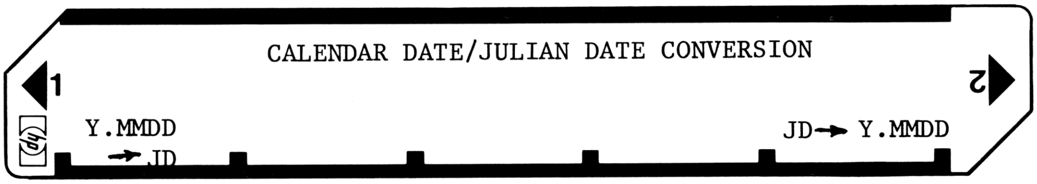
1) Determine starting Julian date(see companion pages)
2) Load card 1.
3) 2429870 $\boxed{\text{ENTER}\uparrow}$ → 2429870.00
4) 12783 $\boxed{+}$ → 2442653.00
5) E --- 1975.0828 ← Answer: Aug.28,1975

Reference(s)

Henry F. Fliegel and Thomas C. Van Flandern, "A Machine Algorithm for Processing Calendar Dates," Communications of the ACM, Volume II, Number 10, October, 1968, Page 657.

This is a combination of HP-65 Users' Library Programs 03874A and 03875A by Rayner K. Rosich.

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1.	Enter Program.				
2.	Enter year, month, and day.	YYYY.MMDD		A	
3.	Enter Julian Date	JD (days)	E		YYYY.MMDD

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	Strip off YYYY	057	1	01	
002	ENT↑	-21	of YYYY.MMDD	058	2	02	
003	INT	16 34		059	÷	-24	
004	STO3	35 03	Year --- Reg. 3	060	INT	16 34	
005	-	-45	(YYYY.MMDD-YYYY.)	061	STO6	35 06	
006	EEX	-23		062	RCL4	36 04	
007	2	02	.MMDD---MM.DD	063	4	04	
008	x	-35		064	8	08	
009	ENT↑	-21	Strip off MM	065	0	00	Term 3 =
010	INT	16 34	of MM.DD	066	0	00	
011	STO2	35 02	Month---Reg. 2	067	+	-55	+(1461x(year +
012	-	-45	(MM.DD-MM.)	068	RCL3	36 03	4800+(mo.-14)/12))
013	EEX	-23		069	+	-55	/4
014	2	02		070	1	01	
015	x	-35	.DD---DD.	071	4	04	
016	STO1	35 01	Day--- Reg. 1	072	6	06	→Reg. 7
017	RCL2	36 02		073	1	01	
018	1	01		074	x	-35	
019	4	04		075	4	04	
020	-	-45		076	÷	-24	
021	1	01	$\frac{\text{Month-14}}{12}$---Reg.4	077	INT	16 34	
022	2	02		078	STO7	35 07	
023	÷	-24		079	3	03	
024	INT	16 34		080	2	02	
025	STO4	35 04		081	0	00	Julian Date =
026	4	04	Term 1 =	082	7	07	
027	9	09		083	5	05	Day -32075
028	0	00	-(3X((Year	084	-	-45	+Term 3 + Term 2
029	0	00	+ 4900	085	RCL1	36 01	+ Term 1
030	+	-55	+(Mo.-14)/12)/10	086	+	-55	
031	RCL3	36 03	))/4	087	RCL6	36 06	↳ x → Display
032	+	-55		088	+	-55	
033	EEX	-23	↳Reg.5	089	RCL5	36 05	
034	2	02		090	+	-55	Set display to
035	÷	-24		091	DSP0	-63 00	YYYY.MMDD
036	INT	16 34		092	RTN	24	for mat.
037	3	03		093	*LBLE	21 15	
038	x	-35		094	DSP4	-63 04	
039	4	04		095	ENT↑	-21	
040	÷	-24		096	6	06	
041	INT	16 34		097	8	08	L = Julian Date
042	CHS	-22		098	5	05	+68569
043	STO5	35 05		099	6	06	
044	RCL4	36 04		100	9	09	→ Reg. 4
045	1	01		101	+	-55	
046	2	02		102	STO4	35 04	
047	x	-35		103	4	04	
048	CHS	-22	Term 2 =	104	x	-35	N = 4xL/146097
049	2	02		105	1	01	
050	-	-45	+ (367x(Month-2	106	4	04	→ Reg. 5
051	RCL2	36 02	-((Month-14)/12)	107	6	06	
052	+	-55	x12))/12	108	0	00	
053	3	03		109	9	09	
054	6	06	↳Reg. 6	110	7	07	
055	7	07		111	÷	-24	
056	x	-35		112	INT	16 34	

REGISTERS

0	1 Day	2 Month	3 Year	4 (mo.-14)/12	5 Term 1	6 Term 2	7 Term 3	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I

97 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	ST05	35 05	
114	1	01	
115	4	04	
116	6	06	L = L
117	0	00	
118	9	09	-(146097xN+3)/4
119	7	07	
120	x	-35	→ Reg. 4
121	3	03	
122	+	-55	
123	4	04	
124	÷	-24	
125	INT	16 34	
126	CHS	-22	
127	RCL4	36 04	
128	+	-55	
129	ST04	35 04	
130	1	01	
131	+	-55	
132	4	04	
133	0	00	
134	0	00	
135	0	00	Year =
136	x	-35	
137	1	01	I=4000x(L+1)/146100
138	4	04	
139	6	06	→ Reg. 3
140	1	01	
141	0	00	
142	0	00	
143	1	01	
144	÷	-24	
145	INT	16 34	
146	ST03	35 03	
147	1	01	
148	4	04	
149	6	06	
150	1	01	
151	x	-35	
152	4	04	L = L
153	÷	-24	
154	INT	16 34	- 1461 x I/4
155	CHS	-22	+ 31
156	3	03	
157	1	01	→ Reg. 4
158	+	-55	
159	RCL4	36 04	
160	+	-55	
161	ST04	35 04	
162	8	08	Month =
163	0	00	
164	x	-35	J = 80xL/2447
165	2	02	→ Reg. 2
166	4	04	
167	4	04	
168	7	07	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	÷	-24	
170	INT	16 34	
171	ST02	35 02	
172	2	02	
173	4	04	
174	4	04	
175	7	07	Day =
176	x	-35	K = L-2447xJ/80
177	8	08	
178	0	00	→ Reg. 1
179	÷	-24	
180	INT	16 34	→ Display at
181	CHS	-22	Stop.
182	RCL4	36 04	
183	+	-55	
184	ST01	35 01	
185	RCL2	36 02	L = J/11
186	1	01	
187	1	01	Reg. 4
188	÷	-24	
189	INT	16 34	
190	ST04	35 04	
191	1	01	
192	2	02	
193	x	-35	Month =
194	CHS	-22	
195	2	02	J=J+2-12xL
196	+	-55	
197	RCL2	36 02	Reg. 2
198	+	-55	
199	ST02	35 02	
200	RCL5	36 05	
201	4	04	
202	9	09	Year =
203	-	-45	I = 100x(N-49)
204	EEX	-23	
205	2	02	+ I + L
206	x	-35	Reg. 3
207	RCL3	36 03	
208	+	-55	
209	RCL4	36 04	
210	+	-55	
211	ST03	35 03	
212	RCL2	36 02	
213	EEX	-23	Clear the stack
214	2	02	
215	÷	-24	and put the
216	+	-55	result into "X"
217	RCL1	36 01	
218	EEX	-23	in the form
219	4	04	YYYY.MMDD
220	÷	-24	
221	+	-55	
222	RTN	24	

LABELS					FLAGS
A	B	C	D	E	0
a	b	c	d	e	1
0	1	2	3	4	2
5	6	7	8	9	3

SET STATUS

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

Program Description I

Program Title DAYS TO DATES AND DATES TO DAYS; DAY OF WEEK

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis **State** Oregon **Zip Code** 97330

Program Description, Equations, Variables

This program computes the number of days elapsed since a certain fixed date in antiquity, using the Formula:

$$D(m,d,y)=d+ [365.25\ g(y,m)] - [3/4([\frac{g(y,m)}{100}]- 7)]+[30.6f(m)]$$

where $f(m) = \begin{cases} m+13 & \text{if } m=1 \text{ or } 2 \\ m+1 & \text{if } m > 2 \end{cases}$ $g(y,m) = \begin{cases} y-1 & \text{if } m=1 \text{ or } 2 \\ y & \text{if } m > 2 \end{cases}$ [ ] = f INT

The difference in days between two dates is computed by subtracting the base value D from the current value.

The day of week is computed by taking D mod 7.

The program uses dates coded mm.ddyyyy.

LBLa computes D(m,d,y), which is the number of days since a fixed point in antiquity; then subtracts from the previous D. The Formulas are:

$$D(m,d,y) = [365.25\ g(y,m)] + [30.6f(m)] + d$$

Where $g(y,m) = \begin{cases} y-1 & \text{if } m=1 \text{ or } m=2 \\ y & \text{if } m > 2 \end{cases}$ $f(m) = \begin{cases} m+13 & \text{if } m=1 \text{ or } m=2 \\ m+1 & \text{if } m > 2 \end{cases}$ [ ] = f INT

LBL b takes the date already coded by fA, adds in the days to get x, and decodes as follows:

$$\hat{y}(x) = [\frac{x-122.1}{365.25}] \qquad \hat{m}(x] = [\frac{x-[365.25\ \hat{y}\ (x)]}{30.6001}$$

$$d(x) = x - [365.25\ \hat{y}\ (x)] - [30.6001\ \hat{m}(x)]$$

$$y(x) = g^{-1}\ (\hat{y}(x), \hat{m}(x) \qquad m(x) = f^{-1}\ (\hat{m}(x))$$

Where g^{-1} and f^{-1} are the "inverses" of g and f given above.

Operating Limits and Warnings: Functions under Label A and Label B are valid from the beginning of the Gregorian Calendar through the year 9999.

Error messages are not given for illegal entries.

Shifted functions are valid from March 1, 1900 through February 28, 2100.

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.

Program Description II

Sketch(es)

Sample Problem(s)

1. I was born on May 11, 1949. How old was I on:

March 4, 1958? August 16, 1969? February 23, 1975?

2. What day-of-week was that last date (February 23, 1975)?

3. How many days are between Jan. 1, 1900 and Jan. 1, 1901?

between Jan 1, 2000 and Jan. 1, 2001?

4. What is the day-of-week of: Nov. 11, 1918? July 4, 1776?

Dec. 7, 1941? Nov. 22, 1963?

5. How many days are there from May 11, 1949 to Aug. 16, 1969?

6. What date is 9000 days from May 11, 1949?

7. How many days from Feb. 28, 1964 to March 1, 1964?

8. How many days from Feb. 28, 1965 to March 1, 1965?

9. What date is 2 days from February 28, 1976?

10. What date is 1 day from Dec. 31, 1959?

Solution(s)

```
1.  5.111949 [A]  3.041958   [B]   Answer:  3219 days
                  8.161969   [B]   Answer:  7402 days
                  2.231975   [B]   Answer:   9419 days
                             [C]   Answer:  0=Sunday
2.  1.011900   [A]   1.011901   [B]   Answer:  365 days
3.  1.012000   [A]   1.012001   [B]   Answer:  366 days
4. 11.111918   [C]   Answer:  1 = Monday
   12.071941   [C]   Answer:  0 = Sunday      7.041776   [C]   Answer:  4 = Thursday
   11.221963   [C]   Answer:  5 = Friday
```

(Continued on next page)

Reference(s) The formulae are the original work of the contributor.
The base date format was suggested by Larry Sullins of Anchorage, Alaska.
Formulas are the original work of the contributor and of Andrew Harrington,

(Continued on next page)

Program Description II

Sketch(es)

Sample Problem(s)

Solution(s)

5. 5.111949 [f] [A] 8.161969 [f] [A] → 7402
6. 5.111949 [f] [A] 9000 [f] [B] → 12.311973
7. 2.281964 [f] [A] 3.011964 [f] [A] → 2
8. 2.281965 [f] [A] 3.011965 [f] [A] → 1
9. 2.281976 [f] [A] 2 [f] [B] → 3.011976
10. 12.311959 [f] [A] 1 [f] [B] → 1.011960

Reference(s) This program is a modification of the Users' Library Program Nos.00256B and 00277B, submitted by Eric Isaacson.

User Instructions

1 →DAYS →DATE 2

DAYS TO DATES AND DATES TO DAYS: DAY OF WEEK

BASE SECOND D-O-WEEK

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Enter base date	mm.ddyyyy	A		New base minus
					old base
3	Enter second date	mm.ddyyyy	B		Date minus base
	Repeat step 3 for new date with same base.				
	Go to step 2 for new base.				
	To calculate day-of-week, press [C] either				
	after punching [A] or [B], or directly after				
	inputting date:				
		mm.ddyyyy	A	C	day-of-wk.
	OR	mm.ddyyyy	B	C	day-of-wk.
	OR	mm.ddyyyy	C		day-of-wk.
					⇓
					Coded:
					0 = Sun.
					1 = Mon.
					2 = Tues
					3 = Wed.
					4 = Thurs.
					5 = Fri.
					6 = Sat.
	To compute the number of days:				
4	Enter Date 1	mm.ddyyyy	f	A	
5	Enter Date 2	mm.ddyyyy	f	A	Date 2-Date 1
6	Enter Date 3	mm.ddyyyy	f	A	Date 3-Date 2
7	Enter Date 4	mm.ddyyyy	f	A	Date 4-Date 3
	etc.				
	To compute the date of given number of days				
	from a given date:				
4'	Enter date	mm.ddyyyy	f	A	
	Enter number of days	number of days	f	B	New date
	Go to step 4' for new case				

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	GSBB	23 12	Compute days
003	RCL1	36 01	
004	STO2	35 02	Store base
005	X⇄Y	-41	Return display
006	RTN	24	
007	*LBLB	21 12	
008	9	09	
009	ENT↑	-21	
010	EEX	-23	
011	6	06	
012	CHS	-22	
013	-	-45	
014	X⇄Y	-41	
015	3	03	
016	X⇄Y	-41	
017	X≤Y?	16-35	
018	GSB0	23 00	
019	1	01	
020	+	-55	m→f(m) y→g(y,m)
021	ENT↑	-21	
022	INT	16 34	
023	STO1	35 01	
024	-	-45	,ddyyyy
025	EEX	-23	
026	2	02	
027	×	-35	dd·yyyy
028	ENT↑	-21	
029	FRC	16 44	
030	EEX	-23	
031	2	02	
032	×	-35	yy.yy
033	ENT↑	-21	
034	INT	16 34	C-no. of centuries
035	7	07	
036	-	-45	
037	.	-62	
038	7	07	
039	5	05	
040	×	-35	
041	INT	16 34	[3/4(c-7)]
042	X⇄Y	-41	yy.yy
043	3	03	
044	6	06	
045	5	05	
046	2	02	
047	5	05	
048	×	-35	
049	INT	16 34	[365.25y]
050	-	-45	
051	-	-45	
052	INT	16 34	d+[365.25y]-[3/4
053	RCL1	36 01	(c-7)]
054	3	03	
055	0	00	
056	.	-62	

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	6	06	
058	×	-35	
059	INT	16 34	[30.6f(m)]
060	+	-55	D(m,d,y)
061	STO1	35 01	
062	RCL2	36 02	Base date
063	-	-45	difference
064	RTN	24	
065	*LBLC	21 13	If input is integer
066	ENT↑	-21	use last date
067	INT	16 34	entered; if not
068	X≠Y?	16-32	interpret as a date
069	GSB1	23 01	
070	RCL1	36 01	
071	7	07	
072	÷	-24	
073	FRC	16 44	
074	7	07	
075	×	-35	Days mod 7
076	RTN	24	
077	*LBL0	21 00	
078	+	-55	
079	+	-55	
080	RTN	24	
081	*LBL1	21 01	
082	X⇄Y	-41	
083	GTOB	22 12	
084	*LBLa	21 16 11	
085	STO0	35 00	
086	DSP0	-63 00	
087	EEX	-23	
088	2	02	
089	STO2	35 02	100 → R_2
090	1	01	
091	2	02	
092	2	02	
093	.	-62	
094	1	01	122.1 → R_3
095	STO3	35 03	
096	EEX	-23	
097	6	06	
098	CHS	-22	10^{-6} → R_4
099	STO4	35 04	
100	3	03	
101	0	00	
102	.	-62	
103	6	06	
104	0	00	
105	0	00	
106	1	01	30.6001 → R_5
107	STO5	35 05	
108	EEX	-23	
109	4	04	10000 → R_8
110	STO8	35 08	
111	3	03	
112	6	06	

REGISTERS

0	1 f(m)(in B) cur. date.Code after B	2 Base Date coded No.&100	3 122.1	4 10^{-6}	5 30.6001	6 365.25	7 Used	8 10000	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
A	B	C	D	E	I				

97 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	5	05	
114	.	-62	
115	2	02	
116	5	05	
117	ST06	35 06	365.25 → R_6
118	RCL0	36 00	Dates to days
119	9	09	
120	ENT↑	-21	
121	ENT↑	-21	
122	RCL4	36 04	
123	-	-45	
124	3	03	
125	R↑	16-31	
126	X≤Y?	16-35	If m is Jan or Feb
127	GSB0	23 00	shift it to prev.yr.
128	1	01	
129	+	-55	
130	ENT↑	-21	
131	INT	16 34	Month
132	ST07	35 07	
133	-	-45	
134	RCL2	36 02	dd.yyyy
135	×	-35	
136	ENT↑	-21	
137	FRC	16 44	.yyyy
138	RCL6	36 06	
139	×	-35	
140	RCL8	36 08	
141	×	-35	365.25(yyyy)
142	INT	16 34	year-days
143	+	-55	
144	INT	16 34	year-days + days-
145	RCL7	36 07	of month
146	RCL5	36 05	
147	×	-35	
148	INT	16 34	Month days
149	+	-55	Total days
150	RCL1	36 01	Previous answer
151	X⇄Y	-41	
152	ST01	35 01	
153	-	-45	
154	CHS	-22	Days since prev. date
155	RTN	24	
156	*LBLb	21 16 12	Days to date
157	RCL1	36 01	
158	+	-55	Code number x
159	ENT↑	-21	
160	ENT↑	-21	
161	RCL3	36 03	
162	-	-45	
163	RCL6	36 06	
164	÷	-24	
165	INT	16 34	
166	ENT↑	-21	
167	ENT↑	-21	
168	RCL4	36 04	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	×	-35	
170	ST01	35 01	
171	CLX	-51	
172	RCL6	36 06	
173	×	-35	
174	INT	16 34	
175	-	-45	
176	ENT↑	-21	
177	ENT↑	-21	
178	RCL5	36 05	
179	÷	-24	
180	INT	16 34	
181	ST07	35 07	
182	RCL5	36 05	
183	×	-35	
184	INT	16 34	
185	-	-45	
186	RCL2	36 02	
187	÷	-24	.dd
188	RCL1	36 01	.ddyyyy
189	+	-55	
190	RCL4	36 04	
191	RCL7	36 07	
192	1	01	
193	-	-45	
194	1	01	
195	2	02	Shift Jan & Feb.
196	X⇄Y	-41	back to proper yr
197	X>Y?	16-34	
198	GSB2	23 02	
199	R↑	16-31	mm.ddyyyy
200	+	-55	
201	DSP6	-63 06	
202	RTN	24	
203	*LBL2	21 02	
204	-	-45	
205	-	-45	
206	RTN	24	
207	R/S	51	
210			
220			

LABELS

A Base	B Second	C D-O-Wk	D	E
a →Days	b →Date	c	d	e
0 Used	1 Used	2 Used	3	4
5	6	7	8	9

FLAGS

0
1
2
3

SET STATUS

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

Program Description I

Program Title Day of Year - Day of Week FAL-12

Contributor's Name F.A. Lummus, P.E.

Address Route 2 Box 84

City Greenville State Texas Zip Code 75401

Program Description, Equations, Variables Given a date, this program primarily calculates the day of year (DOY). The remaining number of days in the year is available. Also, given a year and a DOY, this program will calculate the date. The day of week (DOW) is available. The DOY and date are calculated using the Julian Day Formulas in the HP standard Pac.

Operating Limits and Warnings This program is good for the 200-year period from 1901 through 2099. There are no checks for invalid inputs, however there is a check for data constants storage.

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.

Program Description II

Sketch(es)

Sample Problem(s) (1) Find the day of year of July 4, 1976.

(2) Find the remaining days in the year from the above date.

(3) Find the day of the week for the above date.

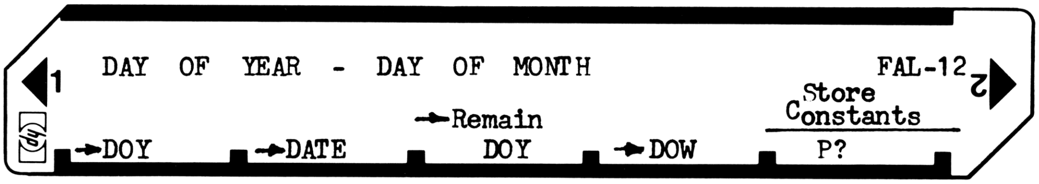
(4) Find the date for the 320th day of 1957.

(5) Find the day of the week for the above date.

```
Solution(s)  (1)   7.04 1976   [A]    →    7.04 1976  ***
                                                186.  ***
             (2)               [C]    →         180.  ***
             (3)               [D]    →           0.  ***  (Sunday)
             (4)        1957   [ENT]  →        1957.
                         320   [B]    →         320.  ***
                                               1957.  ***
                                           11.161957  ***
             (5)               [D]    →           6.  ***  (Saturday)
```

Reference(s) HP-97 Standard Pac.

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Load Side 1 and Side 2				
2	Initialize Constants		f	E	0.00
3	Select Print Option 1.00 yes/o.00 no		E		1.00/0.00
4	For Day of Year in Put Date	Date	A		DOY
	(mm.ddyyyy)				
5	For date, enter year & DOY	Year	ENT		Year
		DOY	B		Date
	——After Either Step 4 or Step 5——				
6	For Remaining Days in the Year		C		Days
7	For Day of Week of Date		D		DOW
	DOW Coded: O=Sun, 1=Mon, etc.				

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	RCLA	36 11	Input Date
003	X=0?	16-43	
004	÷	-24	
005	R↓	-31	
006	DSP6	-63 06	
007	F0?	16 23 00	
008	PRTX	-14	
009	GSBa	23 16 11	Compute Dn
010	STO2	35 02	
011	RCL4	36 04	
012	EEX	-23	
013	6	06	
014	÷	-24	
015	1	01	
016	+	-55	
017	GSBa	23 16 11	Compute Dn of
018	RCL2	36 02	first of year
019	-	-45	
020	CHS	-22	
021	DSP0	-63 00	
022	STO0	35 00	
023	F0?	16 23 00	
024	PRTX	-14	DOY
025	RTN	24	
026	*LBLa	21 16 11	
027	ENT↑	-21	Break input
028	INT	16 34	into individual
029	STO7	35 07	components and
030	-	-45	compute day
031	EEX	-23	number
032	2	02	
033	×	-35	
034	ENT↑	-21	
035	INT	16 34	
036	STO8	35 08	
037	-	-45	
038	EEX	-23	
039	4	04	
040	×	-35	
041	STO9	35 09	
042	STO4	35 04	
043	RCL7	36 07	
044	1	01	
045	+	-55	
046	ENT↑	-21	
047	1/X	52	
048	.	-62	
049	7	07	
050	+	-55	
051	CHS	-22	
052	GSB4	23 04	
053	RCLB	36 12	
054	×	-35	
055	INT	16 34	
056	RCL9	36 09	

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	RCLA	36 11	
058	×	-35	
059	INT	16 34	
060	+	-55	
061	RCL8	36 08	
062	+	-55	
063	RTN	24	
064	*LBL4	21 04	
065	INT	16 34	Correct
066	ST+9	35-55 09	M & Y
067	1	01	
068	2	02	
069	×	-35	
070	-	-45	
071	RTN	24	
072	*LBLB	21 12	
073	RCLA	36 11	Input
074	X=0?	16-43	Year & DOY
075	÷	-24	
076	R↓	-31	
077	STO0	35 00	
078	DSP0	-63 00	
079	F0?	16 23 00	
080	PRTX	-14	
081	R↓	-31	
082	F0?	16 23 00	
083	PRTX	-14	
084	STO4	35 04	
085	DSP6	-63 06	
086	EEX	-23	
087	6	06	
088	÷	-24	
089	1	01	
090	+	-55	Compute DN of
091	GSBa	23 16 11	First of Year
092	RCL0	36 00	Add DOY
093	+	-55	
094	STO2	35 02	Decipher DN
095	RCLC	36 13	into display
096	-	-45	mm.ddyyyy
097	RCLA	36 11	
098	÷	-24	
099	INT	16 34	
100	STO9	35 09	
101	RCLA	36 11	
102	×	-35	
103	INT	16 34	
104	RCL2	36 02	
105	-	-45	
106	CHS	-22	
107	STO6	35 06	
108	RCLB	36 12	
109	÷	-24	
110	INT	16 34	
111	STO7	35 07	
112	RCL6	36 06	

REGISTERS

0 DOY	1	2 Julian Day No.	3 365.5	4 Year	5	6 Used	7 Month	8 Day	9 Year
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A 365.25	B 30.6001	C 122.1	D	E	I

Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	X⇄Y	-41	
114	RCLB	36 12	
115	×	-35	
116	INT	16 34	
117	-	-45	
118	STO8	35 08	
119	RCL7	36 07	
120	1	01	
121	RCL8	36 08	
122	%	55	
123	-	-45	
124	-	-45	
125	RCL7	36 07	
126	1	01	
127	4	04	
128	÷	-24	
129	GSB4	23 04	
130	RCL9	36 09	
131	EEX	-23	
132	6	06	
133	÷	-24	
134	+	-55	
135	F0?	16 23 00	
136	PRTX	-14	
137	RTN	24	
138	*LBLC	21 13	Compute remaining days in year
139	RCL4	36 04	
140	4	04	
141	÷	-24	
142	FRC	16 44	
143	X=0?	16-43	
144	SF2	16 21 02	
145	RCL3	36 03	
146	.	-62	
147	5	05	
148	F2?	16 23 02	
149	CHS	-22	
150	-	-45	
151	RCL0	36 00	
152	-	-45	
153	DSP0	-63 00	
154	F0?	16 23 00	
155	PRTX	-14	
156	RTN	24	
157	*LBLD	21 14	Compute Day of week
158	RCL2	36 02	
159	5	05	
160	+	-55	
161	7	07	
162	÷	-24	
163	FRC	16 44	
164	7	07	
165	×	-35	
166	DSP0	-63 00	
167	RND	16 24	
168	F0?	16 23 00	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	PRTX	-14	
170	RTN	24	
171	*LBLe	21 16 15	Store data Constants
172	3	03	
173	6	06	
174	5	05	
175	.	-62	
176	2	02	
177	5	05	
178	STOA	35 11	
179	3	03	
180	0	00	
181	.	-62	
182	6	06	
183	0	00	
184	0	00	
185	1	01	
186	STOB	35 12	
187	1	01	
188	2	02	
189	2	02	
190	.	-62	
191	1	01	
192	STOC	35 13	
193	3	03	
194	6	06	
195	5	05	
196	.	-62	
197	5	05	
198	STO3	35 03	
199	CLX	-51	
200	RTN	24	
201	*LBLE	21 15	Print Option
202	DSP2	-63 02	
203	F0?	16 23 00	
204	GTO8	22 08	
205	SF0	16 21 00	
206	1	01	
207	RTN	24	
208	*LBL8	21 08	
209	CF0	16 22 00	
210	CLX	-51	
211	RTN	24	
220			

LABELS

A → DOY	B → DATE	C → Remain DOY	D → DOY	E Print
a Compute DN	b	c	d	e Store Data
0	1	2	3	4 Correct M & Y
5	6	7	8 Print-No	9

FLAGS

0	Print
1	———
2	Leap Yr.
3	———

SET STATUS

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

Program Description I

Program Title NUMBER OF WEEKDAYS BETWEEN TWO DATES

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis **State** Oregon **Zip Code** 97330

Program Description, Equations, Variables Program uses dates coded mm.ddyyyy. Program computes the number of weekdays since a certain fixed date in antiquity, using the formula:

$$W\ (m,d,y) = 5\ [\frac{D(m,d,y)}{7}]\ +\ 1/2\ [\ 1.801\ (D(m,d,y)\ 7)]$$

$$\text{where}\quad D(m,d,y) = d - [3/4([\frac{g(x,m)}{100}] - 7)] + [365.25\ g(y,m)] + [30.6\ f(m)]$$

$$f(m) = \begin{matrix} m + 13 \text{ if } m=1,2 \\ m + 1 \text{ if } m > 2 \end{matrix} \qquad g(y,m) = \left[\begin{matrix} y-1 \text{ if } m=1,2 \\ y \text{ if } m > 2 \end{matrix}\right. \qquad [\ ] \equiv f\ int$$

The program then calculates the difference between the current W and the previous W.

Important Note: These calculations are based on noon-to-noon, so the difference between a weekday and a non-weekday will have a half-day in it. For example, the number of weekdays between Friday and Saturday is 1/2 (half of Friday); the number of weekdays between Sunday and Tuesday is 1 1/2 (all of Monday, half of Tuesday).

Operating Limits and Warnings Valid from the beginning of the Gregorian calendar through the year 9999.

Error messages are not given for illegal entries.

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.

Program Description II

Sketch(es)

Sample Problem(s) How many weekdays are there between:

		Coded:	
1.	Sept. 2, 1963 and June 5, 1964	9,021963	6.051964
2.	Apr. 10, 1974 and Apr. 16, 1974	4.101974	4.161974
3.	Apr. 12, 1974 and Apr. 14, 1974	4.121974	4.141974
4.	Apr. 1, 1974 and May 1, 1974	4.011974	5.011974
5.	Apr. 1, 1974 and Apr. 1, 1975	4.011974	4.011975
6.	Jan 1, 1860 and Jan 1, 1960	1.011860	1.011960

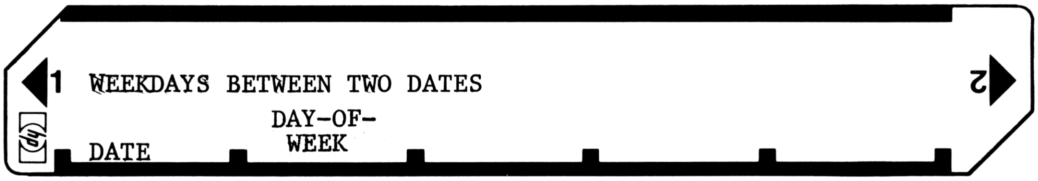
7. What is the day-of-week of the last date entered, Jan. 1, 1960?

Solution(s)

```
1.  9.021963[A] 6.051964[A] --- 199.Ø
2.  4.101974[A] 4.161974[A] ---   4.Ø
3.  4.121974[A] 4.141974[A] ---   Ø.5
4.  4.011974[A] 5.011974[A] ---  22.Ø
5.  4.011974[A] 4.011975[A] --- 261.Ø
6.  1.011860[A] 1.011960[A] --- 26089.5
7.  [B] --- 5 (Friday)
```

Reference(s) This program is a translation of the HP-65 Users' Library Program No. 00282A submitted by Eric Isaacson.

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter Program				
2	Enter first date	mm.ddyyyy	A		(Ignore 1st output)
3	Enter subsequent date	mm.ddyyyy	A		Weekdays between date and previous date
	To show the day-of-week of any date,				
	press B after A		B		day-of-wk of last date entered (0=Sun ... 6=Sat)
	For other dates, do Step 3 repeatedly.				
	Output is always the difference (in				
	weekdays) between the last two dates				
	entered.				

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	Ent.date & compute.
002	9	09	
003	ENT↑	-21	
004	EEX	-23	
005	6	06	
006	CHS	-22	
007	-	-45	
008	X⇄Y	-41	
009	3	03	
010	X⇄Y	-41	
011	X≤Y?	16-35	
012	GSB1	23 01	If M=1or2 add 12 to
013	1	01	M & subt. 1 from y.
014	+	-55	Add 1 to M
015	ENT↑	-21	
016	INT	16 34	
017	STO1	35 01	Store adjusted M.
018	-	-45	.ddyyyy
019	EEX	-23	
020	2	02	
021	×	-35	dd.yyyy
022	ENT↑	-21	
023	FRC	16 44	
024	EEX	-23	
025	2	02	
026	×	-35	yy.yy
027	ENT↑	-21	
028	INT	16 34	C = no. of centuries
029	7	07	
030	-	-45	
031	.	-62	
032	7	07	
033	5	05	
034	×	-35	[3/4
035	INT	16 34	Century days=(C-7)]
036	X⇄Y	-41	yy.yy
037	3	03	
038	6	06	
039	5	05	
040	2	02	
041	5	05	
042	×	-35	
043	INT	16 34	
044	-	-45	
045	-	-45	
046	INT	16 34	[365.25y] + d-
047	RCL1	36 01	[3/4(c-7)]
048	3	03	
049	0	00	
050	.	-62	
051	6	06	
052	×	-35	30.6 f(m)
053	INT	16 34	
054	+	-55	D
055	7	07	
056	÷	-24	D/7

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	ENT↑	-21	
058	FRC	16 44	
059	-	-45	[D/7]
060	LSTX	16-63	1/7(D mod7)
061	STO3	35 03	
062	1	01	
063	2	02	
064	.	-62	
065	6	06	
066	1	01	
067	×	-35	1.801(D mod7)
068	INT	16 34	
069	.	-62	
070	1	01	
071	×	-35	
072	+	-55	
073	5	05	
074	×	-35	W
075	RCL2	36 02	Previous W
076	X⇄Y	-41	
077	STO2	35 02	
078	-	-45	
079	CHS	-22	Difference
080	DSP1	-63 01	
081	RTN	24	
082	*LBLB	21 12	Show day-of-week
083	RCL3	36 03	
084	7	07	
085	×	-35	
086	DSP0	-63 00	
087	RTN	24	
088	*LBL1	21 01	
089	+	-55	
090	+	-55	
091	RTN	24	
092	R/S	51	
100			
110			

REGISTERS

0	1 f(m) (adj.mo.)	2 Code no. prev.date	3 1/7(day-of-wk)	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I

Program Description I

Program Title IN WHAT YEAR IS A GIVEN DATE AN M-DAY?

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis **State** Oregon **Zip Code** 97330

Program Description, Equations, Variables Given a month m, a day-of-month d, a day-of-week w (coded 0=Sun,..., 6=Sat), and a starting year y, the program applies the following formula:

$$f(m) = \begin{cases} m+11 & \text{if } m=1 \text{ or } 2 \\ m+23 & \text{if } m > 2 \end{cases} \qquad g(m) = \begin{cases} 1 & \text{if } m=1 \text{ or } 2 \\ 0 & \text{if } m > 2 \end{cases}$$

$$N(m,d,w) = ([2.6f(m)] + d - w) \mod 7$$

$$h(N) = \begin{cases} 4([\frac{N}{2}] + 4) & \text{if } N \text{ odd} \\ 2N & \text{if } N \text{ even} \end{cases} \qquad [\ ] = f \text{ int}$$

$$L(y,n) = (y-1-h(n) \mod 28 + h(n) + g(m)$$

Then L is a leap year, preceding the specified year, in which the given date is an M-day. By successively adding 6,11,6,5,6,11,6,5,....one obtains all the years one wants.

Operating Limits and Warnings 1. Valid from March 1, 1900 through Feb. 28, 2100.

2. This program does not give error messages for illegal entries.

3. For Feb. 29 non-leap-years as well as leap years will be returned. You should ignore the non-leap years until you get a leap year. Then you can successively add 28 to obtain all the answers.

4. The stack must not be disrupted between steps.

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.

Program Description II

Sketch(es)

Sample Problem(s)

1. When starting in 1949 is May 11 a Sunday?
2. When starting in 1969 is Aug. 16 a Saturday?
3. When starting in 2000 is Christmas first a Sunday?
4. When starting in 1964 is Feb. 14 a Thursday?
5. When starting in 1927 is July 14 a Tuesday?

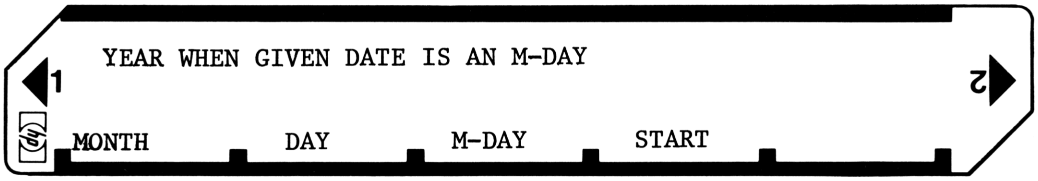
Solution(s)

1. 1952, 1958, 1969, 1975, 1980, etc.
2. 1969, 1975, 1980, 1986, 1997, etc.
3. 2005
4. 1974, 1980, 1985, 1991, 2002, etc.
5. 1931, 1936, 1942, 1953, 1959, etc.

Reference(s) The formulae are the original work of the contributor.

This program is a translation of the HP-65 Users' Library Program No. 00280A submitted by Eric Isaacson, Math Dept., Stanford, California 94305

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Enter month	m	A		
3	Enter Day-of-Month	d	B		
4	Enter Day-of-Week (0=Sun,..., 6=Sat.)	w	C		
5	Enter starting year	y	D		1st correc yr. ≥ y
6	Repeat this step as desired until year except				
	2099			R/S	next cor- rect yr.
				R/S	next cor- rect yr.
				″	″
				″	″
				″	etc.
	Go back to Step 2 for A new case.				
	WARNING: Do not disrupt the stack between				
	steps.				

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11		057	1	01	Enter starting yr.
002	0	00		058	-	-45	Sub 1 for strict inequality
003	STO1	35 01	Enter month	059	ENT↑	-21	
004	CLX	-51		060	R↑	16-31	
005	1	01		061	-	-45	
006	1	01		062	2	02	
007	+	-55	Day-of-week correction factor	063	8	08	
008	1	01		064	÷	-24	
009	4	04	If mo.is Jan or Feb add 12 to mo. & record yr. correction in R_1.	065	INT	16 34	
010	X>Y?	16-34		066	2	02	
011	GTO0	22 00		067	8	08	
012	CLX	-51		068	×	-35	Multiple of 28 in the past.
013	*LBL1	21 01		069	R↑	16-31	
014	2	02		070	RCL1	36 01	
015	.	-62		071	+	-55	Ready to enter addition lop
016	6	06		072	*LBL2	21 02	
017	×	-35	[2.6 m]	073	+	-55	Addition loop
018	INT	16 34		074	X>Y?	16-34	Answer:Is it in range?
019	RTN	24		075	GSB5	23 05	
020	*LBL0	21 00		076	6	06	
021	1	01		077	+	-55	Answer: Is it in range?
022	STO1	35 01		078	X>Y?	16-34	
023	ENT↑	-21		079	GSB5	23 05	
024	+	-55		080	1	01	
025	-	-45		081	1	01	
026	+	-55		082	+	-55	Answer: Is it in range?
027	GTO1	22 01	Enter day-of-mo. [2.6 m] + d	083	X>Y?	16-34	
028	*LBLB	21 12		084	GSB5	23 05	
029	+	-55		085	6	06	
030	RTN	24	Enter day-of-week (M-day) [2.6 m] + d-w	086	+	-55	Answer: Is it in range?
031	*LBLC	21 13		087	X>Y?	16-34	
032	-	-45		088	GSB5	23 05	
033	ENT↑	-21		089	5	05	
034	ENT↑	-21		090	GTO2	22 02	
035	7	07		091	*LBL5	21 05	
036	÷	-24		092	PRTX	-14	
037	INT	16 34		093	R/S	51	
038	7	07	N=([2.6m]+d-w)mod 7	094	RTN	24	
039	×	-35		095	R/S	51	
040	-	-45					
041	2	02	N/2				
042	÷	-24					
043	ENT↑	-21	[N/2]				
044	INT	16 34	If N odd add 4	100			
045	X≠Y?	16-32					
046	GSB0	23 00					
047	4	04					
048	×	-35	Year number mod 28				
049	ENT↑	-21					
050	ENT↑	-21					
051	RTN	24					
052	*LBL0	21 00					
053	4	04					
054	+	-55		110			
055	RTN	24					
056	*LBLD	21 14					

REGISTERS

0	1	2	3	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I

Program Description I

Program Title NUMBER OF M-DAYS BETWEEN TWO DATES AND THE NTH M-DAY OF The Month

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis **State** Oregon **Zip Code** 97330

Program Description, Equations, Variables Program uses dates coded mm.ddyyyy.

Program computes the number of M-days since a certain date in antiquity, using the formula:

$$N(m,d,y,w) = \left[\frac{D(m,d,y,w)}{7}\right] + 1/2\ [0.11\ (D(m,d,y,w) \bmod 7) + 0.9]$$

where $D(m,d,y,w) = d - [3/4([\frac{g(y,m)}{100}] - 7)] + [365.25\ g(y,m)] + [30.6 f(m)] - w$

$$f(m) = \begin{cases} m+13 & m=1 \text{ or } 2 \\ m+1 & m > 2 \end{cases} \qquad g(y,m) = \begin{cases} y-1 & m=1 \text{ or } 2 \\ y & m > 2 \end{cases} \qquad [\ ] = \text{f int}$$

W = M-Day (0 = Sunday, ..., 6 = Saturday)

The program then calculates the difference between the current N and the previous N.

IMPORTANT NOTE: These calculations are based on noon-to-noon, so the difference between an M-Day and a non M-Day will have a half day in it. For example, if M-Day is Tuesday, the difference between a Tuesday and the Wednesday 8 days following is 1 1/2 (half of the first Tuesday plus all of the following Tuesday.)

(Continued on next page)

Operating Limits and Warnings Valid from the beginning of the Gregorian calendar through the year 9999.

Warning: Error messages are not given for illegal dates.

This program does not detect how long a month is -- in particular, it does not detect when the fifth M-Day of a given month does not exist. It tells what it would be if it did exist.

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.

Program Description I

Program Title NUMBER OF M-DAYS BETWEEN TWO DATES AND THE NTH M-DAY OF The Month

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis **State** Oregon **Zip Code** 97330

Program Description, Equations, Variables Program first computes the first M-Day of the given month; given month = m year = y M-day = W (coded 0 = Sunday, ..., 6 = Saturday), the function computed is:

$$D\ (m,y,w) = h\ (W - [2.6\ f(m)] - [5/4\ g(y,m)] + [3/4([\ \frac{g(y,m)}{100} - 7)])$$

$$\text{where } f(m) = \begin{cases} m+13 & \text{if } m=1,2 \\ m+1 & \text{if } m > 2 \end{cases} \qquad g(y,m) = \begin{cases} y-1 & m=1,2 \\ y & m > 2 \end{cases} \qquad h(x) = \begin{cases} 7 \text{ if } & x \bmod 7 = 0 \\ X \bmod 7 & x \bmod 7 \neq 0 \end{cases}$$

$[\quad] = f\ int$

Output is coded in the form mm.ddyyyy.

For the NTH M-Day the appropriate multiple of seven is added to the day-of-month.

Operating Limits and Warnings

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.

Program Description II

Sketch(es)

Sample Problem(s)

1. How many Tuesdays are there between the following pairs of dates?

Codes: Tuesday=2

a. May 10, 1949 and April 2, 1974 -- 5.101949 and 4.021974

b. May 11, 1949 and April 2, 1974 -- 5.111949 and 4.021974

c. May 11, 1949 and April 3, 1974 -- 5.111949 and 4.031974

2. How many Fridays are there between the following pairs of dates?

Friday = 5

a. July 4, 1776 and April 15, 1984-- 7.041776 and 4.151984

b. Feb. 14,1870 and Mar.4, 1958 -- 2.141870 and 3.041958

c. Mar. 23, 1974 and Mar. 28, 1974-- 3.231974 and 3.281974

3. What is the second Wednesday of May, 1949?

4. What is the fifth Friday of March, 1974?

5. What is the first Sunday of January, 2000? (Continued on next page)

Solution(s)

```
1.a.  2 [A] 5.101949[B] 4.021974[B]  →  1299.Ø
  b.  5.111949 [B] 4.021974 [B]      →  1298.5
  c.  5.111949 [B] 4.031974 [B]      →  1299.Ø
2a.   5 [A] 7.041776 [B] 4.151984 [B]→  10842.Ø
 b.   2.141870 [B] 3.041958 [B]      →  4594.Ø
 c.   3.231974 [B] 3.281974 [B]      →     Ø.Ø
3.  5 [f] [A] 1949 [f] [B] 3 [f] [C] + →5.111949 (= May 11, 1949)
4.  3 [f] [A] 1974 [f] [B] 5 [f] [C] 5 [f] [D] → 3.291974 (= Mar. 29,1974
```

(Continued on next page)

Reference(s) The formulas are the original work of the contributor.

This program is a modification of the Users' Library Program Nos. 00281A &00380B, submitted by Eric Isaacson.

Program Description II

Sketch(es)

Sample Problem(s) What is:

6. The second Monday of June 1811?
7. The third Tuesday of August 2525?

Solution(s)

```
5.  1 [f] [A] 2000 [f] [B] 0 [f] [C] → 1.022000 (Jan. 2, 2000)
6.  6 [f] [A] 1811 [f] [B] 1 [f] [C] + → 6.101811 (= June 10, 1811)
7.  8 [f] [A] 2525 [f] [B] 2 [f] [C] 3 [f] [D] → 8.212525 (Aug.21, 2525)
```

Reference(s)

User Instructions

1 NUMBER OF M-DAYS BETWEEN 2 DATES/NTH M-DAY OF A GIVEN MONTH 2

MONTH	YEAR	M-DAY	N	
M-DAY	DATE			

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program		□	□	
2	Enter Day-ofWeek (0=Sun., 1=Mon.,...,6=Sat.)	Day-of-Wk	A		
3	Enter first date	mm.ddyyyy	B		(ignore 1st output)
4	Enter subsequent date	mm.ddyyyy	B		M-Days Between
					previous date.
	For other dates, do Step 4 repeatedly.				
	Output is always the difference in M-Days				
	between the last two dates entered..				
	To change M-Day, go back to Step 2.				
5	Enter month	Month	f	A	
6	Enter year	Year	f	B	
7	Enter Day-of-Week (0=Sun.,...,6-Sat.)	Day-of-wk	f	C	1st M-Day of month
8	To find further M-Days EITHER:				
	Single-step through the M-Days		+		2nd M-Day
			+		3rd M-Day
			+		4th M-Day
			+		5th M-Day
	OR:				
	Enter N	N	f	D	NTH M-Day of Month
					↑↑
	For a new case go back to Step 5.				ALL
					OUTPUT
					CODED
					mm.ddyyyy
	NOTE: In Step 8 you can call f D after				
	single-stepping if you want to				
	(exercising both options), but you				
	cannot single-step after calling f D.				

01006D

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	
002	STO4	35 04	Enter Day-of-Week
003	RTN	24	
004	*LBLE	21 12	Enter date & compute
005	9	09	
006	ENT↑	-21	
007	EEX	-23	
008	6	06	
009	CHS	-22	
010	-	-45	
011	X⇄Y	-41	
012	3	03	
013	X⇄Y	-41	
014	X≤Y?	16-35	If M=1 or 2 add 12
015	GSB9	23 09	to M & sub. 1 from y
016	1	01	Add 1 to M
017	+	-55	
018	ENT↑	-21	
019	INT	16 34	
020	STO1	35 01	Store adjusted M
021	-	-45	.ddyyyy
022	EEX	-23	
023	2	02	
024	x	-35	dd.yyyy
025	ENT↑	-21	
026	FRC	16 44	
027	EEX	-23	
028	2	02	
029	x	-35	yy.yy
030	ENT↑	-21	
031	INT	16 34	C= no. of centuries
032	7	07	
033	-	-45	
034	.	-62	
035	7	07	
036	5	05	
037	x	-35	
038	INT	16 34	Century Days
039	X⇄Y	-41	yy.yy
040	3	03	
041	6	06	
042	5	05	
043	2	02	
044	5	05	
045	x	-35	
046	INT	16 34	[365.25y)
047	-	-45	
048	-	-45	
049	INT	16 34	1+[365.25y]-[3/4
050	RCL1	36 01	(C-7)]
051	3	03	
052	0	00	
053	.	-62	
054	6	06	
055	x	-35	30.6 f(m)
056	INT	16 34	→ Pg 32

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	ST+2	35-55 02	
114	RTN	24	M-Day
115	*LBLc	21 16 13	Adjust so M-day= 0 mod 7
116	RCL1	36 01	
117	2	02	N/7
118	.	-62	
119	6	06	
120	x	-35	[N/7]
121	INT	16 34	
122	-	-45	
123	5	05	
124	RCL2	36 02	
125	.	-62	0.11(N mod 7)
126	0	00	
127	7	07	
128	R↓	-31	
129	x	-35	[0.11(N mod 7)+0.9]
130	4	04	
131	÷	-24	
132	INT	16 34	[$\frac{N}{7}$+1/2[0.11(Nmod 7)+0.9]
133	-	-45	Previous value
134	RCL2	36 02	
135	EEX	-23	
136	2	02	
137	÷	-24	Dif.from prev.value
138	INT	16 34	
139	7	07	
140	-	-45	
141	.	-62	
142	7	07	
143	5	05	
144	x	-35	
145	INT	16 34	
146	+	-55	
147	7	07	Enter month.
148	÷	-24	Clear register 2
149	FRC	16 44	
150	x	-35	
151	+	-55	
152	RCL4	36 04	
153	EEX	-23	Add 1 to m
154	6	06	
155	÷	-24	
156	+	-55	
157	RCL3	36 03	
158	+	-55	If m is Jan. or Feb
159	STO3	35 03	add 12 to m sub-
160	DSP6	-63 06	tract 1 from y.
161	RTN	24	
162	*LBLd	21 16 14	
163	1	01	
164	-	-45	
165	.	-62	f(m)
166	0	00	Enter year Real Year
167	7	07	
168	x	-35	Pg 32

REGIS

0	1 f(m) adjus.mo.	2 Code no. prev.date	3 m	4 Day-of-wk(M-Day)	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
A	B	C	D	E	I				

97 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	+	-55	
058	RCL4	36 04	Adjusted year
059	-	-45	Enter Day-of-Week
060	7	07	(M-Day)
061	÷	-24	
062	ENT↑	-21	
063	FRC	16 44	
064	-	-45	
065	LSTX	16-63	[2.6 f(m)]
066	.	-62	W - [2.6 f(m)]
067	7	07	
068	7	07	
069	×	-35	
070	.	-62	Stack will fill
071	9	09	with .07's
072	+	-55	
073	INT	16 34	
074	2	02	
075	÷	-24	
076	+	-55	
077	RCL2	36 02	W - Month days-
078	X⇄Y	-41	Year days
079	STO2	35 02	
080	-	-45	
081	CHS	-22	
082	DSP1	-63 01	
083	RTN	24	
084	*LBL9	21 09	
085	+	-55	
086	+	-55	
087	RTN	24	
088	*LBL2	21 02	
089	R↓	-31	Century days
090	GTO1	22 01	W-M-days - y days
091	*LBLa	21 16 11	+ C days
092	STO3	35 03	
093	0	00	
094	STO2	35 02	
095	CLX	-51	D(m,y,w)/100
096	1	01	
097	+	-55	Real year
098	4	04	
099	X≤Y?	16-35	
100	GTO2	22 02	.ddyyyy
101	1	01	
102	CHS	-22	mm.ddyyyy
103	STO2	35 02	
104	CLX	-51	
105	8	08	
106	+	-55	
107	+	-55	Enter N
108	*LBL1	21 01	
109	STO1	35 01	N-1
110	RTN	24	
111	*LBLb	21 16 12	Pg 31
112	STO4	35 04	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	RCL3	36 03	Add (N-1) weeks to first M-day.
170	+	-55	
171	RTN	24	
172	R/S	51	

LABELS

A Ent.M-day	B Enter date	C	D	E
a Month	b Year	c Day-of-Wk	d N	e
0	1 Used	2	3	4
5	6	7	8	9

FLAGS

0
1
2
3

SET STATUS

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

Program Description I

Program Title Holidays

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis **State** Oregon **Zip Code** 97330

Program Description, Equations, Variables Use the formulae:

Mother's Day = May 14 - D(0,y)

Father's Day = June 21 - D(3,y)

Election Day = Nov. 8 - D(1,y)

Thanksgiving = Nov. 28 - D(5,y)

Wash.'s Birthday = Feb. 21 - D(2,y-1) y = given year

Labor Day = Sep. 7 - D(3,y)

Columbus Day = Oct. 14 - D(5,y) [ ] = f INT

Veteran's Day = Oct. 28 - D(5,y)

Where $D(x,y) = (x + [\frac{5}{4} y] - [3/4(1+[\frac{y}{100}])])$ mod 7

Outputs the dates coded in the form mm.ddyyyy.

Operating Limits and Warnings

1. Valid from the beginning of the Gregorian calendar through the year 9999.
2. Error messages are not given for illegal entries.
3. Uses current definition of holidays involved for all years.

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.

Program Description II

Sketch(es)

Sample Problem(s)

When is:

1. Mothers Day 2199?
2. Mother's Day 2200?
3. Father's Day 1974?
4. Father's Day 1975?
5. Election Day 1976?
6. Election Day 1977?
7. Thanksgiving 1621?
8. Thanksgiving 1974?
9. Washington's Birthday 1776?
10. Washington's Birthday 1974?
11. Labor Day 1935?
12. Labor Day 1984?
13. Columbus Day 1492?
14. Columbus Day 1992?
15. Veteran's Day 1918?
16. Veteran's Day 1978?

Solution(s)

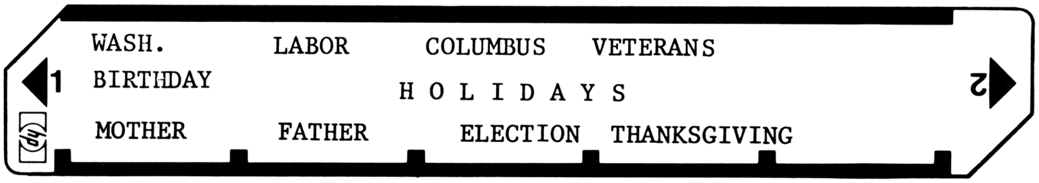
1. 2199[A] ---5.122199 (= May 12, 2199)
2. 2200[A] ---5.112200 (=May 11, 2200)
3. 1974[B] ---6.161974 (= June 16, 1974)
4. 1975[B] ---6.151975 (=June 15, 1975)
5. 1976[C] --11.021976 (=Nov. 2, 1976)
6. 1977[C]---11.081977 (=Nov. 8, 1977)
7. 1621[D]---11.251621 (=Nov. 25, 1621)
8. 1974[D]---11.281974 (=Nov. 28, 1974)
9. 1776[f] [A]---2.191776 (=Feb.19,1776)
10. 1974[f] [A]---2.181974
11. 1935[f] [B]---9.021935
12. 1984[f] [B]---9.031984
13. 1492[f] [C]--10.101492 (Columbus was two days late.)*
14. 1992[f] [C]--10.121992
15. 1918[f] [D]--10.281918 (The Armistice was a couple of weeks late)
16. 1978[f] [D]--10.231978

*But that's okay, Pope Gregory wasn't born yet.

Reference(s) The formulae are the original work of the contributor.

This program is a modification of the Users' Library Program No.s 00278A and 00279A, submitted by Eric Isaacson.

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program				
2	Key in year; choose a holiday	year	A		Mothers Day
			B		Fathers Day
			C		Election Day
			D		Thanks-giving
			f	A	Washington's Birthday
			f	B	Labor Day
			f	C	Columbus Day
			f	D	Veteran's Day

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	Mother's Day
002	0	00	
003	GSBE	23 15	
004	5	05	
005	.	-62	
006	1	01	
007	4	04	
008	+	-55	
009	RTN	24	
010	*LBLB	21 12	Father's Day
011	3	03	
012	GSBE	23 15	
013	6	06	
014	.	-62	
015	2	02	
016	1	01	
017	+	-55	
018	RTN	24	
019	*LBLC	21 13	Election Day
020	1	01	
021	GSBE	23 15	
022	1	01	
023	1	01	
024	.	-62	
025	0	00	
026	8	08	
027	+	-55	
028	RTN	24	
029	*LBLD	21 14	Thanksgiving
030	5	05	
031	GSBE	23 15	
032	1	01	
033	1	01	
034	.	-62	
035	2	02	
036	8	08	
037	+	-55	
038	RTN	24	–
039	*LBLE	21 15	Main Computation Subroutine
040	X⇄Y	-41	
041	ENT↑	-21	
042	ENT↑	-21	
043	R↓	-31	
044	R↓	-31	
045	5	05	
046	×	-35	
047	4	04	
048	÷	-24	
049	INT	16 34	
050	+	-55	
051	X⇄Y	-41	
052	EEX	-23	
053	2	02	
054	÷	-24	
055	INT	16 34	C = Number of Centuries
056	1	01	
057	+	-55	Century year adjustment
058	.	-62	
059	7	07	
060	5	05	
061	×	-35	
062	INT	16 34	
063	-	-45	
064	7	07	
065	÷	-24	
066	FRC	16 44	
067	.	-62	
068	0	00	
069	7	07	
070	×	-35	
071	CHS	-22	Days to be subtracted
072	X⇄Y	-41	
073	EEX	-23	
074	6	06	Encode year
075	÷	-24	
076	+	-55	
077	DSP6	-63 06	
078	RTN	24	
079	*LBLa	21 16 11	Washington's Birthday
080	1	01	
081	-	-45	
082	2	02	
083	GSBE	23 15	
084	2	02	
085	.	-62	
086	2	02	
087	1	01	
088	0	00	
089	0	00	
090	0	00	
091	1	01	
092	+	-55	
093	RTN	24	
094	*LBLb	21 16 12	Labor Day
095	3	03	
096	GSBE	23 15	
097	9	09	
098	.	-62	
099	0	00	
100	7	07	
101	+	-55	
102	RTN	24	
103	*LBLc	21 16 13	Columbus Day
104	5	05	
105	GSBE	23 15	
106	1	01	
107	0	00	
108	.	-62	
109	1	01	
110	4	04	
111	+	-55	
112	RTN	24	

REGISTERS

0	1	2	3	4	5	6	7	8	9
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I

97 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	*LBLd	21 16 14	Veteran's Day
114	5	05	
115	GSBE	23 15	
116	1	01	
117	0	00	
118	.	-62	
119	2	02	
120	8	08	
121	+	-55	
122	RTN	24	
123	R/S	51	
130			
140			
150			
160			

STEP	KEY ENTRY	KEY CODE	COMMENTS
170			
180			
190			
200			
210			
220			

LABELS

A Mother's Day	B Father's Day	C Election Day	D Thanks-giving Day	E Compute
a Wash.'s Birthday	b Labor Day	c Columbus Day	d Veteran's Day	e
0	1	2	3	4
5	6	7	8	9

FLAGS

0	
1	
2	
3	

SET STATUS

FLAGS	ON	OFF
0	□	□
1	□	□
2	□	□
3	□	□

TRIG	
DEG	□
GRAD	□
RAD	□

DISP	
FIX	□
SCI	□
ENG	□
n	____

Program Description I

Program Title EASTER-ASH WEDNESDAY-RELIGIOUS HOLIDAYS FAL-10

Contributor's Name Fred A. Lummus, P. E.

Address Route 2 Box 84

City Greenville State Texas Zip Code 75401

Program Description, Equations, Variables Easter falls on the first Sunday following the arbitrary Paschal Full Moon, which does not necessarily coincide with a real or astronomical full moon. The Golden Number is calculated by adding 1 to the remainder obtained by dividing the year by 19. The Paschal Full Moon is determined by applying the Golden Number to the following table:

Golden		Golden		Golden		Golden		Golden	
Number	Date	Number	Date	Number	Date	Number	Date	Number	Date
1	Apr 14	5	Mar 31	9	Apr 16	13	Aor 2	17	Apr 17
2	Apr 3	6	Apr 16	10	Apr 5	14	Mar 22	18	Apr 7
3	Mar 23	7	Apr 8	11	Mar 25	15	Apr 10	19	Mar 27
4	Apr 11	8	Mar 28	12	Apr 13	16	Mar 30		

If the Paschal Full Moon falls on a Sunday, Easter is on the following Sunday. The earliest Easter can fall is March 23rd and the latest is April 25th. Ash Wednesday and the other religious holidays are determined by specific number of delta days from Easter.

Operating Limits and Warnings This program is good for the 200-year period of March 1, 1900 through February 28, 2100.

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.

Program Description II

Sketch(es)

Sample Problem(s) Determine the religious holidays for the year 1976:

To find Easter and Ash Wednesday:

1976 $\boxed{A}$ → 4.181976 *** Easter

$\boxed{B}$ → 3.031976 *** Ash Wednesday

$\boxed{C}$ → 1. *** Golden Number

Solution(s) To find all religious holidays:

1976 $\boxed{D}$ → 3.031976 *** Ash Wednesday

```
3.071976 ***    First Sunday in Lent
4.041976 ***    Passion Sunday
4.111976 ***    Palm Sunday
4.161976 ***    Good Friday

4.181976 ***    Easter

5.231976 ***    Rogation Sunday
5.271976 ***    Ascension Day
6.061976 ***    Whitsunday
6.131976 ***    Trinity Sunday
```

Reference(s)

User Instructions

EASTER - ASH WEDNESDAY - RELIGIOUS HOLIDAYS					FAL-10
1					2
EASTER	ASH WED	GOLDEN NUMBER	RELIGIOUS HOLIDAYS		P?

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Load Side 1 and Side 2				
2	Load Data Tape (or)	365.25	STO	A	
		30.6001	STO	B	
		122.1	STO	C	
		241303.22	STO	0	
		10281807.26	STO	1	
		15042312.01	STO	2	
		20092717.06	STO	3	
3	Select Print Option No 0.00 Yes 1.00		E		0.00/1.00
4	For EASTER, input year	year	A		EASTER
5	For ASH WEDNESDAY (same year)		B		ASH WED.
6	For GOLDEN NUMBER (same year)		C		GOLD NO.
7	For All Holidays, input year	year	D		DATE 1
					DATE 2
	DATE 1 is ASH WEDNESDAY				DATE 3
	DATE 2 is FIRST SUNDAY IN LENT DATE 3 is PASSION SUNDAY				DATE 4
	DATE 4 is PALM SUNDAY				DATE 5
	DATE 5 is GOOD FRIDAY				
					DATE 6
	DATE 6 is EASTER				
	DATE 7 is ROGATION SUNDAY				DATE 7
	DATE 8 is ASCENSION DAY				DATE 8
	DATE 9 is WHITSUNDAY DATE 10 is TRINITY SUNDAY				DATE 9
					DATE 10
8	For new case, go to step 4 or step 7.				

Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	Calculate Easter
002	GSBa	23 16 11	
003	GTOc	22 16 13	
004	*LBLa	21 16 11	
005	STO9	35 09	
006	RCLA	36 11	
007	X=0?	16-43	
008	÷	-24	
009	x	-35	
010	INT	16 34	
011	1	01	
012	4	04	
013	3	03	
014	+	-55	
015	STO4	35 04	
016	RCL9	36 09	Calculate Golden Number
017	1	01	
018	9	09	
019	÷	-24	
020	FRC	16 44	
021	1	01	
022	9	09	
023	x	-35	
024	1	01	
025	+	-55	
026	STO5	35 05	
027	5	05	Select appropriate "packed" register
028	÷	-24	
029	DSP1	-63 01	
030	RND	16 24	
031	STOI	35 46	
032	FRC	16 44	
033	X=0?	16-43	
034	GTO0	22 00	
035	.	-62	
036	2	02	
037	X⇄Y	-41	
038	X=Y?	16-33	
039	GTO1	22 01	
040	.	-62	
041	4	04	
042	X⇄Y	-41	
043	X=Y?	16-33	
044	GTO2	22 02	
045	.	-62	
046	6	06	
047	X=Y?	16-33	
048	GTO3	22 03	
049	RCLi	36 45	
050	1	01	
051	GTO5	22 05	
052	*LBL0	21 00	Decipher Appropriate
053	RCLi	36 45	
054	EEX	-23	
055	8	08	
056	GTO5	22 05	

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	*LBL1	21 01	"packed" register to find Paschal Full Moon
058	RCLi	36 45	
059	EEX	-23	
060	6	06	
061	GTO5	22 05	
062	*LBL2	21 02	
063	RCLi	36 45	
064	EEX	-23	
065	4	04	
066	GTO5	22 05	
067	*LBL3	21 03	
068	RCLi	36 45	
069	EEX	-23	
070	2	02	
071	*LBL5	21 05	
072	÷	-24	
073	FRC	16 44	
074	EEX	-23	
075	2	02	
076	x	-35	
077	INT	16 34	
078	ST+4	35-55 04	
079	RCL4	36 04	Determine following Sunday
080	5	05	
081	+	-55	
082	7	07	
083	÷	-24	
084	FRC	16 44	
085	7	07	
086	x	-35	
087	DSP0	-63 00	
088	RND	16 24	
089	7	07	
090	-	-45	
091	ST-4	35-45 04	
092	RTN	24	
093	*LBLB	21 12	Calculate Ash Wednesday
094	4	04	
095	6	06	
096	ST-4	35-45 04	
097	GTOc	22 16 13	
098	*LBLC	21 13	Display Golden Number
099	RCL5	36 05	
100	DSP0	-63 00	
101	F0?	16 23 00	
102	PRTX	-14	
103	RTN	24	
104	*LBLD	21 14	Determine all Religious Holidays
105	F0?	16 23 00	
106	SPC	16-11	
107	GSBa	23 16 11	
108	4	04	Ash Wednesday
109	6	06	
110	ST-4	35-45 04	
111	GSBc	23 16 13	
112	4	04	

REGIS

0 constants	1 constants	2 constants	3 constants	4 Day No.	5 Gold No.	6 used	7 Month	8 Day	9 Year
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A 365.25	B 30.6001	C 122.1	D	E	I Control

97 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
113	ST+4	35-55 04	First Sunday in	169	INT	16 34	
114	GSBc	23 16 13	Lent	170	-	-45	
115	2	02		171	STO8	35 08	
116	8	08		172	RCL7	36 07	
117	ST+4	35-55 04	Passion Sunday	173	1	01	
118	GSBc	23 16 13		174	RCL8	36 08	
119	7	07		175	%	55	
120	ST+4	35-55 04	Palm Sunday	176	-	-45	
121	GSBc	23 16 13		177	-	-45	
122	5	05		178	RCL7	36 07	
123	ST+4	35-55 04	Good Friday	179	1	01	
124	GSBc	23 16 13		180	4	04	
125	F0?	16 23 00		181	÷	-24	
126	SPC	16-11		182	INT	16 34	
127	2	02	Easter	183	ST+9	35-55 09	
128	ST+4	35-55 04		184	1	01	
129	GSBc	23 16 13		185	2	02	
130	PSE	16 51		186	×	-35	
131	F0?	16 23 00		187	-	-45	
132	SPC	16-11		188	RCL9	36 09	
133	3	03		189	EEX	-23	
134	5	05		190	6	06	
135	ST+4	35-55 04	Rogation Sunday	191	÷	-24	
136	GSBc	23 16 13		192	+	-55	
137	4	04		193	DSP6	-63 06	
138	ST+4	35-55 04	Ascension Day	194	PSE	16 51	
139	GSBc	23 16 13		195	F0?	16 23 00	
140	1	01		196	PRTX	-14	
141	0	00		197	RTN	24	
142	ST+4	35-55 04	Whitsunday	198	*LBLE	21 15	
143	GSBc	23 16 13		199	F0?	16 23 00	Print Option
144	7	07		200	GTO9	22 09	
145	ST+4	35-55 04	Trinity Sunday	201	SF0	16 21 00	
146	*LBLc	21 16 13		202	CLX	-51	
147	RCL4	36 04	Decipher	203	1	01	
148	RCLC	36 13	Day Number	204	RTN	24	
149	-	-45	of Holiday	205	*LBL9	21 09	
150	RCLA	36 11	into Display	206	CF0	16 22 00	
151	÷	-24	mm.ddyyyy	207	CLX	-51	
152	INT	16 34		208	RTN	24	
153	STO9	35 09		210			
154	RCLA	36 11					
155	×	-35					
156	INT	16 34					
157	RCL4	36 04					
158	-	-45					
159	CHS	-22					
160	STO6	35 06					
161	RCLB	36 12					
162	÷	-24					
163	INT	16 34		220			
164	STO7	35 07					
165	RCL6	36 06					
166	X⇄Y	-41					
167	RCLB	36 12					
168	×	-35					

LABELS					FLAGS	SET STATUS		
A Easter	B Ash Wed	C Gold No.	D Holidays	E Print	0 Print	FLAGS	TRIG	DISP
a calc Easter	b	c decipher DN	d	e	1 —	ON OFF 0 □ ☒ 1 □ ☒ 2 □ ☒ 3 □ ☒	DEG ☒ GRAD □ RAD □	FIX ☒ SCI □ ENG □ n 2
0 0	1 0.2	2 0.4	3 0.6	4	2 —			
5 Next Sun	6	7	8	9 Print	3 —			

Program Description I

Program Title Complete Maya Calendar

Contributor's Name Robert Stone and Larry Goynes

Address 2112 Guadalupe #808

City Austin State Texas Zip Code 78705

Program Description, Equations, Variables Program presents all aspects of the Maya calendar in a pocket calculator format. Program description consists of the following sections: (1) Symbols, (2) Equations, and (3) Tables.

(1) Symbols used.

	LC date	CR date			
		SR date		VY date	
	B. K. T. U. K'	C	N	D	M
Typical date as published:	9.12. 2. 0.16	5	Cib	14	Yaxkin
As input/output here:	9.12 02 00 16	5.	14	14	06(00)

Day Name (N) Codes	01 = Akbal	06 = Lamat	11 = Ben	16 = Etz'nab
	02 = Kan	07 = Muluc	12 = Ix	17 = Cauac
	03 = Chicchan	08 = Oc	13 = Men	18 = Ahau
	04 = Cimi	09 = Chuen	14 = Cib	19 = Imix
	05 = Manik	10 = Eb	15 = Caban	20 = Ik

Month Name (M) Codes	00 = Pop	05 = Xul	10 = Zac	15 = Pax
	01 = Uo	06 = Yaxkin	11 = Ceh	16 = Kayab
	02 = Zip	07 = Mol	12 = Mac	17 = Cumku
	03 = Zotz	08 = Ch'en	13 = Kankin	18 = Uayeb*
	04 = Zec	09 = Yax	14 = Muan	*has only 5 days, 0 to 4

Abbreviations.

B = Baktun, 144000 suns
C = Coefficient of N (Gods 1 to 13)
CR = Calendar Round of 18980 days
D = Day of the month (0 to 19)
G = Lords of the Nights (1 to 9)
K = Katun, 7200 suns
K' = K'in, 1 "sun" (day)

LC = Long Count of days
M = "Month" of 20 days
N = Day Name
SR = Sacred Round of 260 days
T = Tun, 360 suns
U = Uinal, 20 suns
VY = Vague Year of 365 days

0 = (Month) "Seated"

Operating Limits and Warnings Program will give reliable answers out to a distance number of ± 69000 Baktuns, about 27 million years in the past or future. With major period end dates, however, it can handle computations beyond these limits. See the initialization routine on page 4 for dates in the so-called "expanded" LC.

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.

Program Description I

Program Title	Complete Maya Calendar
Contributor's Name	Robert Stone and Larry Goynes
Address	2112 Guadalupe #808
City	Austin State Texas Zip Code 78705

Program Description, Equations, Variables (2) Equations. Card 1 (1)-(4); Card 2 (5)-(12)

(1) Base 10 to "Base 20" conversion is performed by modulo arithmetic.
(2) LC in Base 10 = 144000B + 7200K + 360T + 20U + K' = nth Day of the LC
(3) nth D of CR = (LC in Base 10 - 1842) modulo 18980
(4) nth D of VY, M, D, nth D of SR, N, and C are found by the same method as (3).
(5) nth D of SR = 40((C - N) + 13q) modulo 260 + N
(6) nth D of VY = D + 20M
(7) nth D of CR = 365((nth D of SR - (nth D of VY + CR Type - 3)) + 260q)mod 18980
+ (nth D of VY + CR Type - 3)
(8) r = INT((Nearest Earlier LC date - (nth D of CR + 1842))/18980) + 1
(9) LC in Base 10 = nth D of CR + 1842 + 18980r
(10) JDN = LC in Base 10 + Ahau Constant
(11) Moon Age = JDN (modulo 29.530589) - 9 + 29.530589q
(12) Venus Age = JDN (modulo 583.921) - 110 + 583.921q
where in (5), (7), (11), (12) q = 0 if equation has a positive value; and 1, 2, . . . if equation has a negative value in order to make it positive.

(3) Tables.

A. To Change CR Type

Type	Add to D of M
1	2
2	1
3	0
4	-1
5	-2

Program uses Type 3 for computation. Example of change:
Type 3: 9.12.2.0.16 5 Cib 14 Yaxkin
Type 4: 9.12.2.0.16 5 Cib 13 Yaxkin

B. Moon Age

0	New
7	First Q.
15	Full
22	Last Q.

C. Venus Age

0	Inferior Conj.
0-292	Morning Star
146	Near W. Elong.
292	Superior Conj.
438	Near E. Elong.
292-584	Evening Star

D. Some Ahau Equations

482699	Smiley, 1966
487410	Owen, 1973
489138	Makemson, 1946
489384	Spinden, 1924
584283	Thompson B2, 1958
584284	Thompson B3, 1950
674265	Hochleitner, 1970
679108	Escalona-Ramos,1943
774078	Weitzel, 1947

Operating Limits and Warnings Before using card 2, the CR type must be determined. Table E shows the Days (D) of the Month of the VY that coincide with four Day Names (N) of the SR--Akbal, Lamat, Ben, Etz'nab--in the various CRs. Use Table E according to this example: To determine CR Type of 5 Ahau 3 Mac. If Ahau--an N--falls on the 3rd, then--looking at the Day Name (N) Codes, p. 1--the following Akbal (3 days later) falls on 6 Mac, thus for Table E D = 06. Table E shows that 06 coincides with one of the four Ns only in a Type 3 CR. Similarly, 2 Manik 4 Pop is Type 4: then Lamat falls on 5 Pop; D = 05.

E. To Determine CR Type

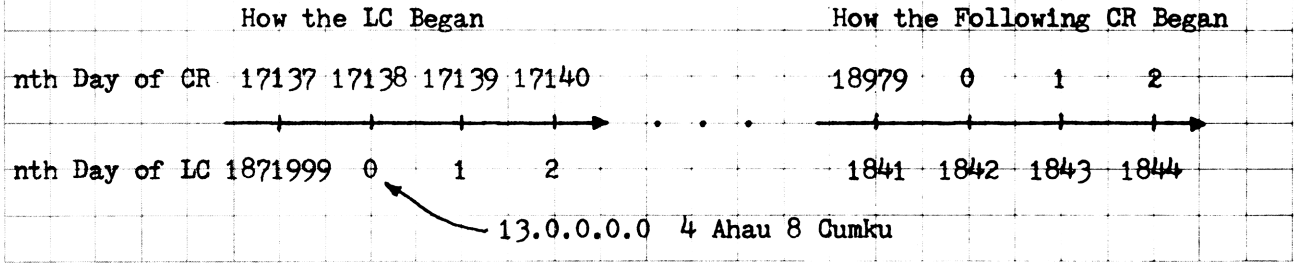
In Type	1	CR,	D =	03, 08, 13, or 18
" "	2	"	" =	02, 07, 12, or 17
" "	3	"	" =	01, 06, 11, or 16
" "	4	"	" =	00, 05, 10, or 15
" "	5	"	" =	04, 09, 14, or 19

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.

Program Description II

Sketch(es) Diagram Showing the Relation of the CR to the LC at its Beginning


How the LC Began
How the Following CR Began
nth Day of CR 17137 17138 17139 17140
18979 0 1 2
nth Day of LC 1871999 0 1 2
1841 1842 1843 1844
13.0.0.0.0 4 Ahau 8 Cumku

Sample Problem(s) (1) What CR date corresponds to 9.12.2.0.16 ?

(2) Show the transition of one CR to the next; include all of the short month Uayeb. Remember that LC in base 10 = nth Day of CR + 1842. See sketch above and Equation (9).

(3) Find the distance number (the number of days between dates) between 6 Imix 19 Zotz and 4 Ahau 13 Mol. Present answer in the Maya way of writing numbers.

(4) What is the 17138th day of the CR? See sketch.

Solution(s) (1) Load card 1, key 9.12020016, A -- 5.14140600 5 Cib 14 Yaxkin

(2) Key 1842, ENTER, 8, -, f, A -- 5.12171700 5 Ix 17 Cumku

10, hSTI, 1, D -- 6.13181700* 7.14191700* 8.15001800* 9.16011800*
10.17021800 * 11.18031800 * 12.19041800 * 13.20000000 * 1.01010000 *
2.02020000*

(3) Load card 2, key 6.191903, A -- 79, 3, B -- 16139

fP⇄S, STO 9, fP⇄S, key 4.181307, A, 3, B -- 12198

RCL E, fP⇄S, RCL 9, fP⇄S, -, + -- 15039 Distance No., Base10

Load card 1, STO 0, E -- 0.02011319 2.1.13.19

(4) (Card 1 loaded), key 17138, C -- 4.18081700 4 Ahau 8 Cumku

Reference(s) Long, R. C. E., "The Highest Known Maya Number", Man, 19 20 (1919), 39-42.

Satterthwaite, Linton, "Calendrics of the Maya Lowlands", Handbook of Middle American Indians, Vol. 3, Univ. of Texas Press, 1965, 603-631.

Thompson, J. Eric S., Maya Hieroglyphic Writing, 3 rd ed., Univ. of Oklahoma Press, 1971.

Program Description II

Sketch(es) Initialization Routine for the Expanded Long Count

Several Maya LC dates reach millions of years--the largest 1.27 billion years--into the past. To check the accuracy of these remarkable dates, convert the distance number to Base 10, divide by 18980 (in R_E) and then repeatedly by 20 (in R_B) until the absolute value of the great number is $\leqslant$400. Take the integer part of this quotient and multiply it by 18980 and by 20 raised to the power the number of times 20 was divided into the giant number. Now subtract this product from the giant number and treat the difference as a usual distance number--see User Instructions, card 1, step 3a.

Sample Problem(s) Quirigua Stela D has the following expanded LC date (Thompson, 316):

```
A ( 0. 1.13. 0.)9.16.15. 0. 0  7 Ahau 18 Pop
B   6. 8.13. 0. 9.16.15. 0. 0  Subtract
C (13)13. 0. 0. 0. 0. 0. 0. 0  7 Ahau  3 Pop
```

Since, in line A, the part not in parentheses is the normal LC date in forward time from the epoch, we may discard its equivalent part in the distance number of line B. The difference, 6.8.13.0.0.0.0.0.0, is the amount of time <u>before</u> the epoch. Make it, therefore, negative.

In Base 10 the distance number is $-(13\text{x}144000\text{x}20^2 + 8\text{x}144000\text{x}20^3 + 6\text{x}144000\text{x}20^4)$ $= -(1.482048 \text{ x } 10^{11})$. Store this number in a safe place.

Solution(s) (1) $-1.482048 \text{ x } 10^{11}/ 18980 = -7808472.08$

(2) Quotients of repeated division by 20 until the absolute value of the last quotient is $\leqslant$ 400: -390423.60, -19521.18, -976.06, -48.80

(3) Note the four divisions by 20. Call this number "E".

(4) Take the integer part of -48.80 and multiply it by 18980 and by 20^E.

$(-48)(18980)(20^4) = -1.457664000 \text{ x } 10^{11}$

(5) Subtract this from the giant distance number.

$-1.482048 \text{ x } 10^{11} - (-1.457664 \text{ x } 10^{11}) = -2438400000$

(6) Store in R_{S9}, load card 1, CLx, A, 1, hSTI, RCL R_{S9}, D -- 7.18030000 (!)

Reference(s)

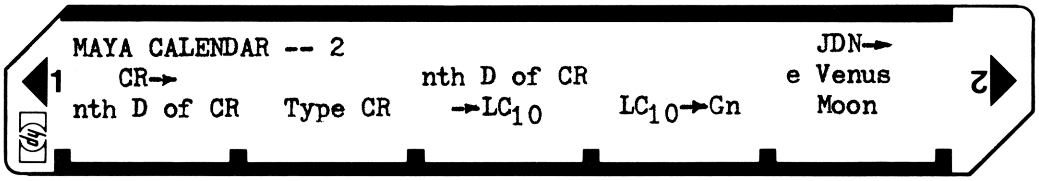
User Instructions

MAYA CALENDAR -- 1

1	a LC_{10}→CR		nth D of CR		(LC_{10} in R_0)	2
	LC_{20}→CR	n_{20}→n_{10}	→CR	Dist. No.	→LC_{20}	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1a	Load card 1.				
2a	Key LC date in "Base 20" form (BB.kkttuuk'k').	BB.kktt...	A		CC.nnddmm
2b	OR: Key LC date as Base 10 integer x.	x	f	A	CC.nnddmm
2c	OR: Key Base 20 number less than 69000 Baktuns	BB.kktt...	B		x Base 10
2d	OR: Key Base 10 number less than 9.9 x 10^9.	x	STO	0	
			E		BB.kktt...
2e	OR: Key nth Day of CR.	nth D ofCR	C		CC.nnddmm
3a	Distance Number function (limit: ±69000 B)				
	After steps 2a or 2b a distance number may be				
	added t number of times.				
4	If distance number is in base 20, go first to				
	step 2c to convert the number to Base 10.				
5	Key the number of times the distance number is				
	to be added.	t	h	ST I	t
6	Key distance number as a Base 10 integer.	± Dist No.	D		CC.nnddmm
7	OPTIONAL: For LC date corresponding to any out-				
	put in step 6, press (E) twice during 5-sec.				
	pause showing CR date. Program will then				
	fetch the LC date of the CR date on pause.				
	After that, program cycles back to step 6 to				
	generate the next CR date in the t series.		E	E	BB.kktt...
3b	Year Bearer function. After steps 2a or 2b,				
	do the following: RCL 0, RCL 8, -, 1, +.		f	A	CC.nn0100
	Read output as follows: "The year is CC.nn."				
2f	Lord of the Katun or Short Count function				
	Given any LC date in Base 20, mentally				
	increase the K by one--if K is 19, make it				
	00 and add 1 to the Baktun instead--and make				
	the T, U, and K' all 00. Key this in.	BB.(kk+1)	A		CC.18ddmm
	C is Lord ("Ahau") of the Katun K+1 that is				
	in the process of being completed.				
3c	Conversion of Maya dates to Christian Era				
	After step 2c, add Ahau Constant (Table D,				
	p. 2) to obtain the Julian Day Number.	Ahau Con.			JDN

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1b	Load card 2.				
2a	Key CR date.	CC.nnddmm	A		nth D ofVY
3	Key type of CR--see Operating Limits and Warn-				
	ings, p. 2.	1 to 5	B		nth D ofCR
4	OPTIONAL: Add or subtract an interval of days,				
	make this sum less than 18980 (available in				
	R_E), load card 1, perform step 2e on p. 5.				
2b	OPTIONAL: Conversion of CR date to LC date.				
	Load card 1, select nearest earlier LC date				
	to within 52 years before the given CR date,				
	convert this to Base 10 by step 2c.				
3	Load card 2, follow steps 2a and 3, and press C		C		LC Base 10
2c	OPTIONAL: The Lord of the Night or Glyph G				
	Given LC in Base 10, press D for the nth lord				
	of the Nine Lords of the Nights.	LC_{10}	D		Glyph G
2d	OPTIONAL: To compute approximate moon age given				
	an LC date. If the date is in Base 10, load				
	card 2; if in Base 20, load card 1 and con-				
	vert the LC date to Base 10 by step 2c, then				
	load card 2.				
3	Key Ahau Constant--see Table D, p. 2.	Ahau Con.	+		JDN
4			E		Moon Age
2e	OPTIONAL: To compute the phase of Venus given				
	an LC date, use the same procedure as step				
	2d except that fE is pressed as the last				
	step.		f	E	Venus Age

67 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	f LBL B	31 25 12	Separate parts of
002	h SF 2	35 51 02	input.
003	GTO 1	22 01	
004	f LBL A	31 25 11	
005	STO 0	33 00	
006	f LBL 1	31 25 01	
007	5	05	
008	h ST I	35 33	
009	2	02	
010	0	00	
011	STO B	33 12	
012	x	71	
013	STO 6	33 06	
014	h x⇄y	35 52	
015	f LBL 2	31 25 02	Indirect STO loop
016	ENTER	41	
017	f INT	31 83	
018	STO (i)	33 24	
019	-	51	
020	RCL 6	34 06	
021	x	71	
022	f DSZ	31 33	
023	GTO 2	22 02	Solve Eq. (2)
024	1	01	# Days in Baktun
025	4	04	
026	4	04	
027	EEX	43	
028	3	03	
029	RCL 5	34 05	#Baktuns
030	x	71	
031	7	07	# Days in Katun
032	2	02	
033	0	00	
034	0	00	
035	RCL 4	34 04	# Katuns
036	x	71	
037	+	61	
038	3	03	# Days in Tun
039	6	06	
040	0	00	
041	RCL 3	34 03	# Tuns
042	x	71	
043	+	61	
044	RCL B	34 12	
045	RCL 2	34 02	# Uinals
046	x	71	
047	+	61	
048	RCL 1	34 01	# K'ins
049	+	61	
050	STO 0	33 00	LC in Base 10
051	h F? 2	35 71 02	
052	R/S	84	--STOP--
053	g LBLf a	32 25 11	
054	STO 0	33 00	Solve Eq. (3).
055	1	01	LC : CR Constant
056	8	08	
057	4	04	
058	2	02	
059	-	51	
060	0	00	
061	h x⇄y	35 52	
062	1	01	# Days in a CR
063	8	08	
064	9	09	
065	8	08	
066	0	00	
067	STO E	33 15	
068	÷	81	
069	gFRAC	32 83	
070	RCL E	34 15	
071	x	71	
072	DSP 0	23 00	
073	f RND	31 24	
074	f x<0	31 71	
075	RCL E	34 15	
076	+	61	
077	f LBL C	31 25 13	Solve Eq. (4)
078	STO 7	33 07	nth D of CR
079	3	03	# Days in a VY
080	6	06	
081	5	05	
082	STO D	33 14	
083	÷	81	
084	g FRAC	32 83	
085	RCL D	34 14	
086	x	71	
087	DSP 0	23 00	
088	f RND	31 24	
089	STO 8	33 08	nth D of VY =
090	2	02	nth D of CR
091	0	00	(mod 365)
092	STO B	33 12	
093	÷	81	
094	ENTER	41	
095	f INT	31 83	
096	STO 3	33 03	M of VY
097	-	51	
098	RCL B	34 12	
099	x	71	
100	STO 2	33 02	D of VY
101	RCL 7	34 07	nth D of SR =
102	2	02	nth D of CR
103	6	06	(mod 260)
104	0	00	
105	STO C	33 13	
106	÷	81	
107	g FRAC	32 83	
108	RCL C	34 13	
109	x	71	
110	f RND	31 24	
111	STO 9	33 09	nth D of SR
112	RCL B	34 12	

REGISTERS

0	1	2	3	4	5	6	7	8	9
LC$_{10}$	K or N	T or D	U or M	K'	B	Dist. #	nth D$_{CR}$	nth D$_{VY}$	nth D$_{SR}$
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
13	20	260	365	18980	Used

67 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
113	÷	81	N = nth D of SR	169	RCL 0	34 00	
114	g FRAC	32 83	(mod 20)	170	h SF 0	35 51 00	
115	RCL B	34 12		171	GTOf a	22 31 11	
116	x	71		172	f LBL 8	31 25 08	
117	f RND	31 24		173	f-x-	31 84	PRINT x
118	f x=0	31 51		174	f DSZ	31 33	
119	RCL B	34 12		175	GTO 7	22 07	
120	STO 1	33 01	N	176	h CF 0	35 61 00	
121	RCL 9	34 09	C = nth D of SR	177	R/S	84	
122	1	01	(mod 13)	178	f LBL E	31 25 15	Base 10 to Base 20
123	3	03		179	RCL 0	34 00	#K' = LC$_{10}$ (mod20)
124	STO A	33 11		180	2	02	
125	÷	81		181	0	00	
126	g FRAC	32 83		182	STO B	33 12	
127	RCL A	34 11		183	÷	81	
128	x	71		184	ENTER	41	
129	f RND	31 24		185	f INT	31 83	
130	f x=0	31 51		186	STO 5	33 05	
131	RCL A	34 11	C	187	-	51	
132	f LBL 4	31 25 04	Collect parts of	188	RCL B	34 12	
133	h F? 2	35 71 02	CR or LC output.	189	x	71	
134	GTO 5	22 05	LC	190	STO 4	33 04	# K'
135	GTO 6	22 06	CR	191	RCL 5	34 05	# U = INT(LC$_{10}$
136	f LBL 5	31 25 05		192	1	01	÷ 20) mod 18
137	RCL 5	34 05	# Baktuns	193	8	08	
138	RCL 4	34 04	# K'ins	194	÷	81	
139	8	08		195	ENTER	41	
140	g 10^x	32 53		196	f INT	31 83	
141	÷	81		197	STO 5	33 05	
142	+	61	BB.000000k'k'	198	-	51	
143	f LBL 6	31 25 06	Call U or M	199	1	01	
144	RCL 3	34 03		200	8	08	
145	6	06		201	x	71	
146	g 10^x	32 53		202	STO 3	33 03	# U
147	÷	81		203	RCL 5	34 05	# T = INT(LC$_{10}$
148	+	61		204	RCL B	34 12	÷ 360) mod 20
149	RCL 2	34 02	Call T or D	205	÷	81	
150	4	04		206	ENTER	41	
151	g 10^x	32 53		207	f INT	31 83	
152	÷	81		208	STO 5	33 05	
153	+	61		209	-	51	
154	RCL 1	34 01	Call K or N	210	RCL B	34 12	
155	2	02		211	x	71	
156	g 10^x	32 53		212	STO 2	33 02	# T
157	÷	81	BB.kkttuuk'k' or	213	RCL 5	34 05	# K = INT(LC$_{10}$
158	+	61	CC.nnddmm00	214	RCL B	34 12	÷ 7200) mod 20
159	DSP 8	23 08		215	÷	81	
160	h F? 0	35 71 00		216	ENTER	41	
161	GTO 8	22 08		217	f INT	31 83	
162	f-x-	31 84	PRINT x	218	STO 5	33 05	# B
163	R/S	84	--STOP--	219	-	51	
164	f LBL D	31 25 14	Distance Number	220	RCL B	34 12	
165	STO 6	33 06		221	x	71	
166	f LBL 7	31 25 07		222	STO 1	33 01	# K
167	RCL 6	34 06		223	h SF 2	35 51 02	
168	STO+0	33 61 00	Add Dist. # to LC.	224	GTO 4	22 04	

LABELS					FLAGS
A Cmput CR	B Cmput LC$_{10}$	C nth D$_{CR}$	D Dist. No.	E Cmput LC$_{20}$	0 Series
a LC$_{10}$ → CR	b	c	d	e	1
0	1 LC Date	2 STO loop	3 nth D$_{LC}$	4 Collect	2 LC vs CR
5 Used	6 Used	7 Add Dist#	8 new LC	9	3

SET STATUS

FLAGS	ON	OFF	TRIG		DISP	
0	□	☒	DEG	☒	FIX	☒
1	□	☒	GRAD	□	SCI	□
2	□	☒	RAD	□	ENG	□
3	□	☒			n 8	

67 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	f LBL A	31 25 11	Separate parts of input.	057	R/S	84	--STOP--
002	4	04		058	f LBL B	31 25 12	Input CR Type
003	h ST I	35 33		059	3	03	
004	2	02		060	-	51	
005	5	05		061	+	61	
006	x	71		062	DSP 0	23 00	
007	STO 5	33 05		063	f RND	31 24	
008	h x⇄y	35 52		064	f x=0	31 51	
009	f LBL 1	31 25 01	Indirect STO loop.	065	GTO 3	22 03	
010	ENTER	41		066	STO 8	33 08	nth D of VY
011	f INT	31 83		067	GTO 4	22 04	
012	STO (i)	33 24		068	f LBL 3	31 25 03	If nth D of VY = 0 make it = 365.
013	-	51		069	RCL D	34 14	
014	RCL 5	34 05		070	STO 8	33 08	
015	x	71		071	f LBL 4	31 25 04	Solve Eq. (7).
016	f DSZ	31 33		072	1	01	
017	GTO 1	22 01		073	8	08	
018	4	04	Solve Eq. (5).	074	9	09	
019	0	00		075	8	08	
020	RCL 4	34 04		076	0	00	
021	RCL 3	34 03		077	STO E	33 15	
022	-	51		078	RCL D	34 14	
023	f x>0	31 81		079	RCL 9	34 09	
024	GTO 2	22 02		080	RCL 8	34 08	
025	1	01		081	-	51	
026	3	03		082	f x>0	31 81	
027	STO A	33 11		083	GTO 6	22 06	
028	+	61		084	f LBL 5	31 25 05	
029	f x>0	31 81		085	RCL C	34 13	
030	GTO 2	22 02		086	+	61	
031	RCL A	34 11		087	f x>0	31 81	
032	+	61		088	GTO 6	22 06	
033	f LBL 2	31 25 02		089	GTO 5	22 05	
034	x	71		090	f LBL 6	31 25 06	
035	2	02		091	x	71	
036	6	06		092	h x⇄y	35 52	
037	0	00		093	÷	81	
038	STO C	33 13		094	g FRAC	32 83	
039	÷	81		095	x	71	
040	g FRAC	32 83		096	RCL 8	34 08	
041	RCL C	34 13		097	+	61	
042	x	71		098	f RND	31 24	
043	RCL 3	34 03		099	STO 7	33 07	nth D of CR
044	+	61		100	DSP 2	23 02	
045	STO 9	33 09	nth D of SR	101	f-x-	31 84	PRINT x
046	3	03		102	h RTN	35 22	--STOP--
047	6	06		103	f LBL C	31 25 13	Compute LC_{10}. Solve Eq. (8).
048	5	05		104	STO 7	33 07	
049	STO D	33 14		105	1	01	
050	RCL 1	34 01	Solve Eq. (6).	106	8	08	
051	2	02		107	4	04	
052	0	00		108	2	02	
053	STO B	33 12		109	STO 6	33 06	
054	x	71		110	+	61	
055	RCL 2	34 02		111	RCL 0	34 00	
056	+	61		112	h x⇄y	35 52	

REGISTERS

0 LC_{10}	1 MM	2 DD	3 NN	4 CC	5 Syn. Mo.	6 Venus Pr	7 nth D_{CR}	8 nth D_{VY}	9 nth D_{SR}
S0 JDN	S1	S2	S3	S4	S5	S6	S7	S8	S9

A 13	B 20	C 260	D 365	E 18980	I Used

67 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
113	-	51		169	x	71	
114	1	01		170	9	09	Adjust to real time.
115	8	08		171	-	51	
116	9	09		172	RCL 5	34 05	
117	8	08		173	h x⇌y	35 52	
118	0	00		174	f x<0	31 71	Make positive.
119	STO E	33 15		175	+	61	
120	÷	81		176	f-x-	31 84	PRINT x
121	ENTER	41		177	h RTN	35 22	--STOP--
122	g FRAC	32 83		178	g LBLf e	31 25 15	Compute Venus age.
123	f x=0	31 51		179	f P⇌S	31 42	
124	GTO 7	22 07		180	STO 0	33 00	
125	h R↓	35 53		181	f P⇌S	31 42	Solve Eq. (12).
126	f INT	31 83		182	5	05	Syn. Per. of Venus
127	1	01		183	8	08	
128	f LBL 7	31 25 07		184	3	03	
129	+	61	r	185	.	83	
130	RCL E	34 15	Solve Eq. (9).	186	9	09	
131	x	71		187	2	02	
132	RCL 6	34 06		188	1	01	
133	RCL 7	34 07		189	STO 6	33 06	
134	+	61		190	÷	81	
135	+	61		191	g FRAC	32 83	
136	STO 0	33 00		192	RCL 6	34 06	
137	DSP 2	23 02		193	x	71	
138	h RTN	35 22	--STOP--	194	1	01	Adjust to real time.
139	f LBL D	31 25 14	Compute glyph G.	195	1	01	
140	9	09	n = LC_{10} mod. 9	196	0	00	
141	÷	81		197	-	51	
142	g FRAC	32 83		198	RCL 6	34 06	
143	9	09		199	h x⇌y	35 52	
144	x	71		200	f x<0	31 71	Make positive.
145	0	00		201	+	61	
146	h x⇌y	35 52		202	f-x-	31 84	PRINT x
147	f x=0	31 51		203	h RTN	35 22	--STOP--
148	9	09					
149	+	61					
150	f-x-	31 84	PRINT x				
151	h RTN	35 22	--STOP--				
152	f LBL E	31 25 15	Compute Moon age.				
153	f P⇌S	31 42					
154	STO 0	33 00		210			
155	f P⇌S	31 42					
156	2	02					
157	9	09					
158	.	83					
159	5	05					
160	3	03					
161	0	00					
162	5	05					
163	8	08					
164	9	09		220			
165	STO 5	33 05					
166	÷	81					
167	g FRAC	32 83					
168	RCL 5	34 05					

LABELS					FLAGS
A Sto input	B Type CR	C CR to LC	D Glyph G	E Moon	0
a	b	c	d	e Venus	1
0	1 STO loop	2 Eq. (5)	3 Last D_{VY}	4 Eq. (7)	2
5 Eq. (7)	6 Eq. (7)	7 Eq. (9)	8	9	3

SET STATUS

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

Program Description I

Program Title MOHAMMEDAN (ISLAM) - GREGORIAN CALENDAR CONVERSION FAL-11

Contributor's Name Fred A. Lummus, P. E.

Address Route 2 Box 84

City Greenville State Texas Zip Code 75401

Program Description, Equations, Variables The Mohammedan (Islam) Calendar is a lunar reckoning from the year of the Hegira, 622 A.D. It runs in cycles of 30 years of which the 2nd, 5th, 7th, 10th, 13th, 16th, 18th, 21st, 24th, 26th, and 29th are leap years, having 355 days; the others are common, having 354 days. The following table gives the months of the Mohammedan Year and the number of days each:

1	Muhharram	30	5	Jumada I	30	9	Ramadan	30
2	Safar	29	6	Jumada II	29	10	Shawwal	29
3	Rabia I	30	7	Rajab	30	11	Zu'lkadah	30
4	Rabia II	29	8	Shaban	29	12	Zu'lhijjah	29*

* In leap years, 30 days

The following formulas are applicable for the 10,631 days of the 44th cycle, therefore all dates are referred to that cycle for calculation.

$$y' = y - 1 \qquad m' = m - 1$$

$$DN_m = \text{RND}\left[(y')(354.3659)\right] + \text{INT}\left[(m')(29.5001) + 0.5\right] + d + \left[\text{INT}\left(\frac{y}{30}\right) - 44\right](10631)$$

$$\text{Julian Day Number}_{J.D._m} = DN_m + 1{,}948{,}440$$

Operating Limits and Warnings This program is good for a 400-year Gregorian period of March 1, 1800 through February 28, 2200 which corresponds to the Mohammedan (Islam) period of Shawwal 4, 1214 (10.021214) through Muharram 13, 1627 (1.131627).

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.

Program Description II

Sketch(es)

Sample Problem(s) Convert the Mohammedan date of Safar 2, 1391 to the corresponding Gregorian date:

2.021391 $\boxed{A}$ → 2.021391 ***
2441041. (Display only) (Julian Day Number)
$\boxed{E}$ → 3.301971 ***

Find the corresponding Mohammedan date for July 4, 1976:

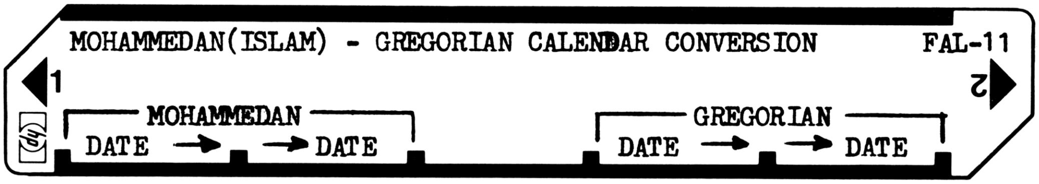
7.041976 $\boxed{D}$ → 7.041976 ***
2442964. (Display only) (Julian Day Number)
$\boxed{B}$ → 7.071396 ***
Rajab 7, 1396

Solution(s) Additional corresponding dates which can be verified:

Zu'lkadah 1,1389	(11.101389)	**and**	January 9, 1970	(1.091970)
Muharram 1, 1390	(1.011390)	and	March 9, 1970	(3.091970)
Rabia I 1, 1390	(3.011390)	and	May 7, 1970	(5.071970)
Jumada II 1, 1390	(6.011390)	and	August 4, 1970	(8.041970)
Safar 1, 1391	(2.011391)	and	March 29, 1971	(3.291971)
Shaban 1, 1391	(8.011391)	and	September 22,1971	(9.221971)
Shawwal 1, 1391	(10.011391)	and	November 20, 1971	(11.201971)
Muharram 1, 1398	(1.011398)	and	December 12, 1977	(12.111977)

Reference(s)

User Instructions



STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Load Side 1 and Side 2				
2	Load Data Tape (or)	365.25	STO	0	
		30.6001	STO	1	
		122.1	STO	2	
		1720982	STO	3	
		694097	STO	4	
		767146	STO	5	
		30	STO	A	
		0.5	STO	B	
		355	STO	C	
		12.3	STO	E	
			f	P S	
		354.3659	STO	0	
		29.5001	STO	1	
		10631	STO	2	
		1948440	STO	3	
			f	P S	
3	For Mohammedan (Islam) Dates:				
	● To Find Julian Day Number(J.D.),	DATE$_m$	A		J.D.
	input Date (mm.ddyyyy)				
	● To Find Mohammedan Date,	J.D.	B		DATE$_m$
	input Julian Day Number				
4	For Gregorian Dates:				
	● To Find Julian Day Number,	DATE$_G$	D		J.D.
	input Date (mm.ddyyyy)				
	● To Find Gregorian Date,	J.D.	E		DATE$_G$
	input Julian Day Number				
5	To Convert Mohammedan Date to Gregorian Date,	DATE$_m$	A		
	input Date (mm.ddyyyy)		E		DATE$_G$
6	To Convert Gregorian Date to Mohammedan Date,	DATE$_G$	D		
	input Date (mm.ddyyyy)		B		DATE$_m$

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLA	21 11	Compute Julian Day Number of Mohammedan Date
002	P⇄S	16-51	
003	GSB0	23 00	
004	GSB5	23 05	
005	GSBb	23 16 12	
006	GSB2	23 02	
007	RCL7	36 07	
008	GSBb	23 16 12	
009	RCL1	36 01	
010	×	-35	
011	RCLB	36 12	
012	+	-55	
013	INT	16 34	
014	+	-55	
015	RCL8	36 08	
016	+	-55	
017	RCL3	36 03	
018	+	-55	
019	P⇄S	16-51	
020	RTN	24	
021	*LBLB	21 12	
022	P⇄S	16-51	Decipher Julian Day Number into Display mm.ddyyyy
023	RCL3	36 03	
024	-	-45	
025	STO4	35 04	
026	RCL0	36 00	
027	÷	-24	
028	INT	16 34	
029	*LBL9	21 09	
030	STO9	35 09	
031	GSB5	23 05	
032	GSB2	23 02	
033	RCL4	36 04	
034	X⇄Y	-41	
035	-	-45	
036	RCLC	36 13	
037	X⇄Y	-41	
038	X=Y?	16-33	
039	GTO6	22 06	
040	X=0?	16-43	
041	GTO8	22 08	
042	STO4	35 04	
043	RCLB	36 12	
044	-	-45	
045	RCL1	36 01	
046	÷	-24	
047	INT	16 34	
048	STO7	35 07	
049	RCL1	36 01	
050	×	-35	
051	RCLB	36 12	
052	+	-55	
053	INT	16 34	
054	RCL4	36 04	
055	-	-45	
056	CHS	-22	

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	EEX	-23	
058	2	02	
059	÷	-24	
060	RCL7	36 07	
061	GSBa	23 16 11	
062	+	-55	
063	*LBL7	21 07	
064	RCL9	36 09	
065	GSBa	23 16 11	
066	P⇄S	16-51	
067	GTO3	22 03	
068	*LBLE	21 15	
069	RCL3	36 03	Decipher Julian Day Number into Display mm.ddyyyy
070	-	-45	
071	RCL5	36 05	
072	X⇄Y	-41	
073	X≤Y?	16-35	
074	GSBb	23 16 12	
075	RCL6	36 06	
076	X⇄Y	-41	
077	X>Y?	16-34	
078	GSBa	23 16 11	
079	STO4	35 04	
080	RCL2	36 02	
081	-	-45	
082	RCL0	36 00	
083	÷	-24	
084	INT	16 34	
085	STO9	35 09	
086	RCL0	36 00	
087	×	-35	
088	INT	16 34	
089	RCL4	36 04	
090	-	-45	
091	CHS	-22	
092	STOD	35 14	
093	RCL1	36 01	
094	÷	-24	
095	INT	16 34	
096	STO7	35 07	
097	RCLD	36 14	
098	X⇄Y	-41	
099	RCL1	36 01	
100	×	-35	
101	INT	16 34	
102	-	-45	
103	STO8	35 08	
104	RCL7	36 07	
105	1	01	
106	RCL8	36 08	
107	%	55	
108	-	-45	
109	-	-45	
110	RCL7	36 07	
111	1	01	
112	4	04	

REGISTERS

0	1	2	3	4	5	6	7	8	9
365.25	30.6001	122.1	1720982	Date-G	694097	767146	Month-G	Day - G	Year-G
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
354.3659	29.5001	10631	1948440	Date-M		used	Month-M	Day - M	Year-M

A	B	C	D	E	I
30	0.5	355	used	12.30	

Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	÷	-24	
114	GSB4	23 04	
115	RCL9	36 09	
116	GTO3	22 03	
117	*LBL0	21 00	Break Date into the Individual Components of mm.ddyyyy
118	PRTX	-14	
119	STO4	35 04	
120	ENT↑	-21	
121	INT	16 34	
122	STO7	35 07	
123	-	-45	
124	EEX	-23	
125	2	02	
126	×	-35	
127	ENT↑	-21	
128	INT	16 34	
129	STO8	35 08	
130	-	-45	
131	EEX	-23	
132	4	04	
133	×	-35	
134	STO9	35 09	
135	RTN	24	
136	*LBLD	21 14	Compute Julian Day Number of Gregorian Date
137	GSB0	23 00	
138	RCL7	36 07	
139	GSBa	23 16 11	
140	ENT↑	-21	
141	1/X	52	
142	.	-62	
143	7	07	
144	+	-55	
145	CHS	-22	
146	GSB4	23 04	
147	RCL1	36 01	
148	×	-35	
149	INT	16 34	
150	RCL9	36 09	
151	RCL0	36 00	
152	×	-35	
153	INT	16 34	
154	+	-55	
155	RCL8	36 08	
156	+	-55	
157	RCL5	36 05	
158	X⇄Y	-41	
159	X≤Y?	16-35	
160	GSBa	23 16 11	
161	RCL6	36 06	
162	X⇄Y	-41	
163	X>Y?	16-34	
164	GSBb	23 16 12	
165	RCL3	36 03	
166	+	-55	
167	RTN	24	
168	*LBL2	21 02	
169	RCL0	36 00	
170	×	-35	
171	DSP0	-63 00	
172	RND	16 24	
173	DSP6	-63 06	
174	RCL6	36 06	
175	RCL2	36 02	
176	×	-35	
177	+	-55	
178	RTN	24	
179	*LBL3	21 03	
180	EEX	-23	
181	6	06	
182	÷	-24	
183	+	-55	
184	PRTX	-14	
185	RTN	24	
186	*LBL4	21 04	Correct m & y
187	INT	16 34	
188	ST+9	35-55 09	
189	1	01	
190	2	02	
191	×	-35	
192	-	-45	
193	RTN	24	
194	*LBL5	21 05	Refer to 44th Cycle
195	RCLA	36 11	
196	÷	-24	
197	INT	16 34	
198	4	04	
199	4	04	
200	-	-45	
201	STO6	35 06	
202	RCLA	36 11	
203	×	-35	
204	RCL9	36 09	
205	X⇄Y	-41	
206	-	-45	
207	RTN	24	
208	*LBL6	21 06	
209	RCLE	36 15	
210	GTO7	22 07	
211	*LBL8	21 08	
212	RCL9	36 09	
213	GSBb	23 16 12	
214	GTO9	22 09	
215	*LBLa	21 16 11	
216	1	01	
217	+	-55	
218	RTN	24	
219	*LBLb	21 16 12	
220	1	01	
221	-	-45	
222	RTN	24	

LABELS

A Date$_M$ →	B → Date$_M$	C	D Date$_G$ →	E → Date$_G$
a 1 +	b 1 -	c	d	e
0 Store Date	1 Compute Greg DN	2 Mon year sub	3 used	4 Correct m & y
5 Chg to Basic Yr	6 12.30 sub	7 Rtn pt 12.30	8 Lp year repeat	9 Rtn pt Lp year

FLAGS

0	—
1	—
2	—
3	—

SET STATUS

FLAGS	ON	OFF
0	□	☒
1	□	☒
2	□	☒
3	□	☒

TRIG	
DEG	☒
GRAD	□
RAD	□

DISP	
FIX	☒
SCI	□
ENG	□
n	6

Program Description I

Program Title CHINESE YEARS TO/FROM GREGORIAN YEARS

Contributor's Name Hewlett-Packard

Address 1000 N.E. Circle Blvd.

City Corvallis State Oregon Zip Code 97330

Program Description, Equations, Variables

The Chinese year designation consists of two characters, the "most significant bit" called (干) and the "least significant bit" called (支). There are 10 (干)'s and 12 (支)'s which are represented here by consecutive numbers for computation purposes. The rule states that any odd number of Group One can associate with only the odd numbers of Group Two, and likewise even numbers can associate with only even numbers. This way, 60 possible combinations form one basic cycle of 60 years. Each cycle begins with the same year (甲子), or (1, 1) in the numeralized notations, or (1.01) in our machine notations, which falls in the years 1924 and 1924 ± 60N; where N is any integer. This program arbitrarily selects (N = -85) as the internal datum but displays 1924 (N = 0) externally as the first year and 1983 as the 60th year of the machine cycle. Actually, the machine cycle may be set to start with a year corresponding to any N. The fact that the sum of the two numbers for any year is always an even number provides us with a parity check. For example, the year 1975 falls in row 2 and column 4 (2+4=6), and the year is (2.04) or (乙卯).

Operating Limits and Warnings

Chinese New Years which start the first months may fall in January or February of the Gregorian scale.

The year (2.10), for example, should be interpreted as TWO-TEN nor TWO-ONE. Notice that 2 + 1 = 3 violates the parity.

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.

CHINESE YEARS TO/FROM GREGORIAN YEARS

ORDER	YYYY	$Y_1Y_1 . Y_2Y_2$	干支
1	1924	1 . 01	(甲 子)
2	1925	2 . 02	(乙 丑)
3	1926	3 . 03	(丙 寅)
4	1927	4 . 04	(丁 卯)
5	1928	5 . 05	(戊 辰)
6	1929	6 . 06	(己 巳)
7	1930	7 . 07	(庚 午)
8	1931	8 . 08	(辛 未)
9	1932	9 . 09	(壬 申)
10	1933	10 . 10	(癸 酉)
11	1934	1 . 11	(甲 戌)
12	1935	2 . 12	(乙 亥)
13	1936	3 . 01	(丙 子)
14	1937	4 . 02	(丁 丑)
...	...	...	...
59	1982	9 . 11	(壬 戌)
60	1983	10 . 12	(癸 亥)

	子 1	丑 2	寅 3	卯 4	辰 5	巳 6	午 7	未 8	申 9	酉 10	戌 11	亥 12
甲 1	1924		1974		1964		1954		1944		1934	
乙 2		1925		1975		1965		1955		1945		1935
丙 3	1936		1926		1976		1966		1956		1946	
丁 4		1937		1927		1977		1967		1957		1947
戊 5	1948		1938		1928		1978		1968		1958	
己 6		1949		1939		1929		1979		1969		1959
庚 7	1960		1950		1940		1930		1980		1970	
辛 8		1961		1951		1941		1931		1981		1971
壬 9	1972		1962		1952		1942		1932		1982	
癸 10		1973		1963		1953		1943		1933		1983

Program Description II

Sketch(es)

Sample Problem(s)

(1) Convert the years 1924 and 1864 to Chinese years.

Key E 1924 A reads 1.01 : row 1& Column 1 (甲子).

Key E 1864 A reads 1.01 : row 1 & Column 1 (甲子).

(2) Convert the Chinese year 1.01 (甲子) to Gregorian year.

Key E 1.01 B reads 1924 (1st year of machine cycle.

C reads 1984, C reads 2044, etc.

or D reads 1864, D reads 1804, etc.

(3) Convert the year 1975 to Chinese year and back.

Key E 1975 A reads 2.04 : row 2 & column 4 (乙卯),

B reads 1975.

(4) Convert the year 1776 to Chinese year and back.

Key E 1776 A reads 3.09 : row 3 & column 9 (丙申),

B reads 1956, D D D reads 1776.

(5) Convert the Chinese year "1.02" illegally to Gregorian year.

Actually row 1 and column 2 (1 + 2 = 3) is an empty space in the Table. There is no such a year called 1.02 or (甲丑). But if we are to carry out the computation illegally:

Key E 1.02 B results in Error indicating an illegal operation.

Reference(s)

This program is a translation of the HP-65 User Library Program No. 01994A, submitted by Tak Y. Lee.

Encyclopedia Americana

Encyclopedia Britannica

User Instructions

CHINESE YEARS TO/FROM GREGORIAN YEARS

1 | G--- C | G----C | +60 | -60 | INIT | 2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter Program				
			–		
2	Initialize				
				E	0.
3	Gregorian to Chinese				
	Enter year YYYY		A		$Y_1Y_1.Y_2Y_2$
	or				
3	Chinese to Greg.				
	Enter $Y_1Y_1.Y_2Y_2$		B		Y Y Y Y
	Add 60 to result		C		Y + 60
	...				
	Sub. 60 from result		D		Y - 60
	...				
(For a new case, go to step 2, but for continuous					
	calculations, step 2 may be omitted)				

LABELS

A G→C	B G←C	C +60	D -60	E
a	b	c	d	e
0	1	2	3	4
5	6	7	8	9

FLAGS
0

SET STATUS

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

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLE	21 15	Initialize & enter machine datum yr.
002	CLRG	16-53	
003	3	03	85x60 before 1924
004	1	01	
005	7	07	
006	6	06	
007	CHS	-22	
008	STO1	35 01	
009	1	01	
010	0	00	Cycle of Y_1Y_1
011	STO2	35 02	
012	1	01	
013	2	02	Cycle of Y_2Y_2
014	STO3	35 03	
015	0	00	
016	RTN	24	
017	*LBLA	21 11	Greg. to Chinese Enter YYYY
018	RCL1	36 01	
019	-	-45	
020	STO5	35 05	Δ Y = YYYY -(-3176)
021	RCL5	36 05	
022	RCL2	36 02	
023	÷	-24	
024	INT	16 34	
025	RCL2	36 02	
026	×	-35	
027	-	-45	
028	1	01	
029	+	-55	The most significant
030	RCL5	36 05	
031	RCL5	36 05	
032	RCL3	36 03	
033	÷	-24	
034	INT	16 34	
035	RCL3	36 03	
036	×	-35	
037	-	-45	
038	1	01	
039	+	-55	The least significant.
040	EEX	-23	
041	2	02	
042	÷	-24	
043	+	-55	
044	DSP2	-63 02	$Y_1Y_1.Y_2Y_2$ Display Chinese yr
045	RTN	24	
046	*LBLB	21 12	Chinese to Greg. Enter $Y_1Y_1.Y_2Y_2$
047	INT	16 34	
048	STO6	35 06	Store Y_1Y_1
049	LSTX	16-63	
050	FRC	16 44	
051	EEX	-23	
052	2	02	
053	×	-35	-Store Y_2Y_2
054	X>Y?	16-34	Is $Y_2Y_2 > Y_1Y_1$
055	GSB0	23 00	
056	STO7	35 07	

STEP	KEY ENTRY	KEY CODE	COMMENTS
057	RCL6	36 06	
058	RCL7	36 07	
059	-	-45	
060	2	02	
061	÷	-24	
062	STO8	35 08	$(Y_1Y_1-Y_2Y_2)/2$
063	FRC	16 44	
064	0	00	Parity check
065	X≠Y?	16-32	
066	÷	-24	
067	RCL8	36 08	Compute years in excess of the datum year
068	RCL3	36 03	
069	×	-35	
070	RCL7	36 07	
071	+	-55	
072	1	01	$(\frac{Y_1Y_1-Y_2Y_2}{2})x12+Y_2Y_2-1$
073	-	-45	
074	1	01	
075	9	09	
076	2	02	
077	4	04	
078	+	-55	
079	DSP0	-63 00	Display YYYY within the machine cycle (1924-1983)
080	RTN	24	
081	*LBLC	21 13	
082	6	06	
083	0	00	Add 60 years
084	+	-55	
085	RTN	24	
086	*LBLD	21 14	
087	6	06	
088	0	00	
089	-	-45	Subtract 60 years.
090	RTN	24	
091	*LBL0	21 00	
092	RCL3	36 03	
093	-	-45	
094	RTN	24	
095	R/S	51	
100			
110			

REGISTERS

0	1	2	3	4	5	6	7	8	9
	-3176	10	12		y	Y_1Y_1	Y_2Y_2	$(Y_1Y_1-Y_2Y_2)/2$	Used
									S9

A	B	C	D	E	I

Program Description I

Program Title Biorhythm – Biological Cycles + Calendar Functions

Contributor's Name Rex H. Shudde

Address 27105 Arriba Way

City Carmel **State** CA **Zip Code** 93921

Program Description, Equations, Variables

1. The Julian Day Number is computed using a modification of the routine given by Fliegel & VanFlandern, Comm ACM, 11 (Oct 1968), 657. For Julian Colender Dates prior to 15 Oct 1582, the Julian Day Number is given by

$$JD_J = \left\lfloor 1461\left(I+4800+\left\lfloor \frac{J-14}{12}\right\rfloor\right) \div 4\right\rfloor + \left\lfloor 30.59\left(J-2-12\left\lfloor\frac{J-14}{12}\right\rfloor\right)\right\rfloor + K - 32113$$

For Gregorian Colender Dates from 15 Oct 1582, the Julian Day Number is given by

$$JD_G = JD_J - \left\lfloor 0.75\left\lfloor\left(I+100+\left\lfloor\frac{J-14}{12}\right\rfloor\right)\div 100\right\rfloor\right\rfloor + 2$$

where I = Year, J = month, K = day, and $\lfloor x \rfloor$ is the integer part of x.

2. Day-of-week number = JD mod 7
3. For a C-day biorhythm cycle, compute:
   a. "No of days into cycle" $L = M \bmod C$ where $M = |JD^{\#}2 - JD^{\#}1|$
   b. "Critical Index" $CI = L/C$
   c. Amplitude = $\sin(360\,CI)$
   d. Slope or rate-of-change of cycle = $\cos(360\,CI)$
4. Correction of Julian Day Number for exact time of events: Replace JD by $JD + T/24 + 0.5$ where T is the time of the event in hours.

Operating Limits and Warnings For B.C. dates, the following convention must be observed:

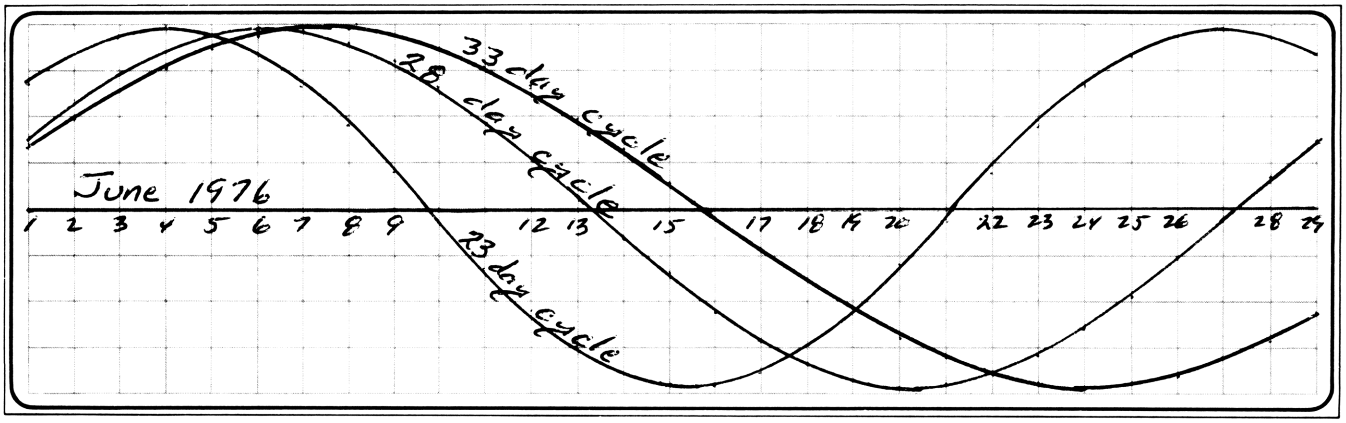
3 B.C = −2
2 B.C = −1
1 B.C = 0
1 A.D = 1
2 A.D = 2

For example, March 27, 935 B.C must be entered as −3.270934 (see step 2a page 3). The preceding − sign must be used for B.C. dates. (From step 2a, obtain JD = 1380000.00 for this sample date).

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.

Program Description II



Sample Problem(s) An individual was born on March 26, 1961 at 1815 hours PST. Find the Julian day number of his birthday and the day of the week. Find the number of days elapsed from his birth until noon PST on June 21, 1976 (also convert this to weeks and days). On June 21, 1976 at noon PST, find how many days of his 23day-, 28day-, and 33 day-cycle have elapsed. Also, calculate the "critical index", the amplitude, and the rate-of-change of each of these cycles. The "critical points" are where the cycle cross the zero line, and are easily determined from the "critical index"" (the fraction of the cycle which as elapsed). A "critical index" of:

a. less than 0.20 is leaving 0 and going toward +

b. between 0.48 and 0.52 is crossing 0 from + toward –

c. greater than 0.98 is entering 0 from –

Also, a critical index between 0.22 and 0.29 indicates a peak+, and between 0.73 and 0.78 is a trough –.

Graphs may be produced by plotting the amplitude of the cycles (which vary from –1.00 to +1.00) on the vertical axis against the date on the horizontal axis. The date can be incremented by pressing $\boxed{f}\boxed{d}$ instead of re-entering each new date.

Soln: 3.261961 $\boxed{A}$ 2437385.00 (Julian Day Number) $\boxed{R/S}$ 6.0 = Sunday. 18.15 $\boxed{f}\boxed{a}$ 0.26 (fraction of day past previous noon). 6.211976 $\boxed{B}$ 2442951.00 (Julian Day Number) $\boxed{R/S}$ 0.00 = Monday. $\boxed{f}\boxed{c}$ 5565.74 = days and fraction of day between the two times. $\boxed{R/S}$ 795.07 = 795 weeks & 0.7 days between dates. 23day cycle: $\boxed{C}$ 22.74 days $\boxed{R/S}$ 0.99 = Crit index (See Graph, and part (c) above) $\boxed{R/S}$ –0.07 = height of curve $\boxed{R/S}$ 1.00 = slope of curve $\boxed{R/S}$ Error = R/S pressed once to often $\boxed{CLX}$ 28day cycle: $\boxed{D}$ 21.74 $\boxed{R/S}$ 0.78 $\boxed{R/S}$ –0.99 $\boxed{R/S}$ 0.17. 33day cycle: $\boxed{E}$ 21.74 $\boxed{R/S}$ 0.66 $\boxed{R/S}$ –0.84, $\boxed{R/S}$ –0.54

Reference(s)

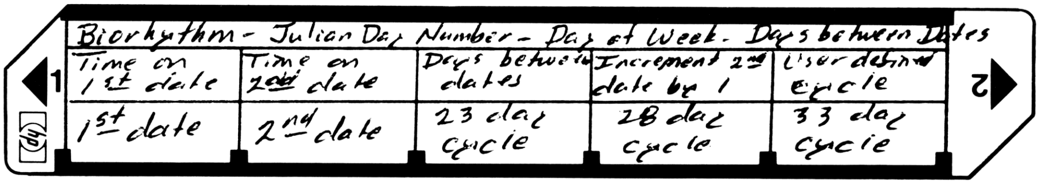
1. Thommen, G.S., "Is This Your Day?", Crown Publ. Inc. NY 1973

2. O'Neil & Phillips, "Biorhythms. How to Live with Your Life Cycles," Ward Ritchie Press, Pasadena 1975.

3. Gittelson, B. "Biorhythm. A Personal Science," Arco Publ. Co., NY 1975

4. Wallerstein, M.R. & Roberts, N.L., Human Behavior, April 1973, pp 8-15.

User Instructions



Biorhythm - Julian Day Number - Day of Week - Days between Dates

1					2
	Time on 1st date	Time on 2nd date	Days between dates	Increment 2nd date by 1	User defined cycle
	1st date	2nd date	23 day cycle	28 day cycle	33 day cycle

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Enter program. Dates are entered in the				
	form MM.DDYYYY where MM is the				
	month number, DD is the 2-digit day, and				
	YYYY is the 4-digit day. Note that a				
	decimal point **must** separate MM and DD.				
2a	Enter 1st date	MM.DDYYYY	A		Jul. Day No.
b	Compute day of week		R/S		0.0 to 6.0
c	Enter time if other than noon (1200 hours)	HH.MMSS	f	A	———
3a	Enter 2nd date	MM.DDYYYY	B		Jul. Day No.
b	Compute day of week		R/S		0.0 to 6.0
c	Enter time if other than noon (1200 hours)	HH.MMSS	f	B	———
4	Any or all of the following steps may be performed.				
5a	Compute days between dates		f	C	DDDD.FF
b	Convert to weeks (WW), days (D), & fraction of day F		R/S		WW.DF
6a	Compute days DD & fraction of days FF of 23 day cycle		C		DD.FF
b	Display critical index		R/S		0.XX
c	Display amplitude of cycle		R/S		±X.XX
d	Display slope of cycle		R/S		±X.XX
7	28 day cycle (see step 6)		D		DD.FF
8	33 day cycle (see step 6)		E		DD.FF
9a	Enter user defined cycle period (eg. 84 days)	84.	f	E	DD.FF
	See 6b,c,d for options				
10a	Increment 2nd date by one day		f	D	Jul. Day No
b	Compute day of week		R/S		0.0 to 6.0
11	Repeat from Step 4 as desired				
	Day of week: 0 = Monday				
	1 = Tuesday				
	etc.				
	* See Warnings pg 1 for B.C. Dates				

67 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	f LBL 9	31 25 09	Set up constants		X	71	
	1	01			RCL 8	34 08	
	2	02			X	71	
	STO A	33 11		060	RCL C	34 13	
	1	01			1	01	
	4	04			4	04	
	6	06			—	51	
	1	01			RCL A	34 11	
	STO 2	33 02			÷	81	
010	4	04			f INT	31 83	
	8	08			STO 9	33 09	
	0	00			+	61	
	0	00			STO 8	33 08	
	STO 3	33 03		070	RCL 3	34 03	
	3	03			+	61	
	2	02			RCL2	34 02	
	1	01			X	71	
	1	01			4	04	
	3	03			÷	81	
020	STO 4	33 04			f INT	31 83	
	2	02			RCL C	34 13	
	2	02			2	02	
	9	09			—	51	
	9	09		080	RCL 9	34 09	
	1	01			RCL A	34 11	
	7	07			X	71	
	0	00			—	51	
	STO 5	33 05			3	03	
	EEX	43			0	00	
030	2	02			•	83	
	STO 6	33 06			5	05	
	EEX	43			9	09	
	4	04			X	71	
	STO 7	33 07		090	f INT	31 83	
	3	03			+	61	
	6	06			RCL D	34 14	
	0	00			+	61	
	h STI	35 33			RCL 4	34 04	
	RTN	35 22			—	51	
040	f LBL 0	31 25 00	Unpack date and generate Julian Day Number		RCL 5	34 05	
	↑	41			h X≗Y	35 52	
	h ABS	35 64			f X≤Y	32 71	
	÷	81			RTN	35 22	Return from Julian Calendar
	STO 8	33 08		100	RCL 8	34 08	
	h LSTX	35 82			RCL 6	34 06	Continue for Gregorian Calendar
	↑	41			+	61	
	f INT	33 83			h LST X	35 82	
	STO C	33 13			÷	81	
	—	51			f INT	31 83	
050	RCL 6	34 06			•	83	
	X	71			7	07	
	↑	41			5	05	
	f INT	31 83			X	71	
	STO D	33 14		110	f INT	31 83	
	—	51			—	51	
	RCL 7	34 07			2	02	

REGISTERS

0 JD#1	1 JD#2	2 1461	3 4800	4 32113	5 2299170	6 100	7 1000	8 I/YYYY	9 $\left\lfloor \frac{J-14}{12} \right\rfloor$
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A 12	B	C J/MM	D K/DD	E Cycle length	I 360

67 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
113	+	61			−	51	
	RTN	3522		170	R/S	84	display
	f LBL 3	31 25 03	Stop and error routine		RCL E	34 15	Critical Index
	R/S	84			÷	81	
	0	00			R/S	84	
	÷	81			h RC I	3534	
	GTO 3	22 03			×	71	
120	f LBL A	31 25 11	Generate first Julian Day Number		1	01	
	STO 0	33 00			f R←	31 72	
	h F? 2	35 71 02			h x⇄y	3552	Amplitude
	f GSB 9	31 22 09			R/S	84	
	RCL 0	34 00		180	h x⇄y	3552	
	f GSB 0	31 22 00			GTO 3	22 03	Slope
	STO 0	33 00			g LBL a	32 25 11	Time entered for 1st date
	R/S	84			f GSB 2	31 22 02	
	GTO 5	22 05			STO + 0	33 61 00	
	f LBL B	31 25 12	Input 2nd date & generate 2nd Julian Day Number		GTO 3	22 03	
130	STO 1	33 01			g LBL b	32 25 12	Time entered for 2nd date
	h F? 2	35 71 02			f GSB 2	31 22 02	
	f GSB 9	31 22 09			STO + 1	33 61 01	
	RCL 1	34 01			GTO 3	23 03	
	f GSB 0	31 22 00		190	f LBL 2	31 25 02	Convert time to proper fraction of a Julian day
	STO 1	33 01			f H←	31 74	
	R/S	84			RCL A	34 11	
	f LBL 5	31 25 05	Set up for day of week		−	51	
	7	07			2	02	
	STO E	33 15			4	04	
140	h R↓	35 53			÷	81	
	GTO 4	22 04			RTN	3522	
	f LBL C	31 25 13	Set up for 23-day cycle		g LBL c	32 25 13	Compute days between dates & display.
	2	02			RCL 0	34 00	
	3	03		200	RCL 1	34 01	
	STO E	33 15			−	51	
	GTO 1	22 01			h ABS	35 64	
	f LBL D	31 25 14	Setup for 28-day cycle		R/S	84	
	2	02			7	07	Compute weeks, days, & fraction of day wwwww.DF
	8	08			÷	81	
150	STO E	33 15			f INT	31 83	
	GTO 1	22 01			h LST X	35 82	
	f LBL E	31 25 15	Setup for 33-day cycle		g FRAC	32 83	
	3	03			·	83	
	3	03		210	7	07	
	STO E	33 15			×	71	
	f LBL 1	31 25 01	Generate # of elapsed days in x-register		+	61	
	RCL 0	34 00			GTO 3	22 03	
	RCL 1	34 01			g LBL d	32 25 14	Increment 2nd Julian Day Number
	−	51			RCL 1	34 01	
160	h ABS	35 64			1	01	
	f LBL 4	31 25 04	Compute # of days into cycle		+	61	
	↑	41			STO 1	33 01	
	↑	41			R/S	84	
	RCL E	34 15		220	GTO 5	22 05	
	÷	81			g LBL e	32 25 15	Set up for user cycle
	f INT	31 83			STO E	33 15	
	RCL E	34 15			GTO 1	22 01	
	×	71			R/S	84	

LABELS

A ✓	B ✓	C ✓	D ✓	E ✓
a ✓	b ✓	c ✓	d ✓	e ✓
0 ✓	1 ✓	2 ✓	3 ✓	4 —
5 ✓	6	7	8	9 ✓

FLAGS

0	
1	
2	✓
3	

SET STATUS

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

Program Description I

Program Title NEW MOON AND FULL MOON DAY OF MONTH FAL-50

Contributor's Name Fred A. Lummus, P.E.

Address Route 2 Box 84

City Greenville **State** Texas **Zip Code** 75401

Program Description, Equations, Variables This program follows an empirical algorithm published in a recent issue of the amateur astronomers monthly, SKY & TELESCOPE. The author of the article, O.L. Harvey claims, "By testing hundreds of dates across the interval from 1001 B.C. to A.D. 2003, I found that about half the answers obtained with this device agree exactly with the day of new moon, and about half are off by one day. A very small number (two percent in my sample) are two days in error." Mr. Harvey developed a three part table based on repeating patterns in an enormous data source and on averages of date differences. An entry point was calculated for the first part, the resultant was the entry point for the second part. A calculated difference on the second part produced a number. The month provided the key for the third part of the table. These numbers are combined with other calculations to determine the day of month. Mr. Vanderburgh devised a HP-65 program to calculate the entry points and when used with the table and additional inputs would complete the calculations for the day of month.

This program uniquely incorporates that table into the registers and selects the appropriate numbers to perform the complete calculations for day of month from a single input. The only restriction is due to the input format of the month and year, all dates in B.C. had to be deleted.

Operating Limits and Warnings Input interger years only. Output range is from o-30; thus 0 indicates that new moon occurs on the last day of the previous month; the 30th of February is either March 1st or 2nd. The Julian Calender applies to dates prior to 1582; the Gregorian thereafter. Dates are limited to A.D. only.

Out of over 300 dates tested, 62% were correct, 37% were off by one day, and less than 1% were two days in error.

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.

Program Description II

Sketch(es)

R		L			M	
9		9	19	29	0	March
25	17	28	8	18	2	April
10	3	17	27	7	2	May
26	19	6	16	26	4	June
12	4	25	5	15	4	July
28	20	14	24	4	6	August
13	6	3	13	23	7	September
29	22	22	2	12	8	October
15	7	11	21	1	9	November
1	23	30	10	20	10	December
					11	January
					12	February

Sample Problem(s) (1) Find the date of the New Moon in:

July 1974

February 1963

May 1970

(2) Find the date of the Full Moon in:

July 1974

March 1963

November 1966

Solution(s)

```
(1)   7.1974  [A] →   7.1974    ***
                      7.191974  ***  Date is correct.
      2.1963  [A] →   2.1963    ***
                      2.231963  ***  Correct date is February 24.
      5.1970  [A] →   5.1970    ***
                      5.051970  ***  Date is correct.
(2)   7.1974  [B] →   7.1974    ***
                      7.041974  ***  Date is correct.
      3.1963  [B] →   3.1963    ***
                      3.101963  ***  Date is correct.
     11.1969  [B] →  11.1969    ***
                     11.241969  ***  Correct date is November 23.
```

Reference(s) Harvey, O.L., A THUMBNAIL ALMANAC FOR THE MOON, Sky & Telescope, Vol. 47, No. 6, June 1974, p. 384.

Vanderburgh, Richard G., "New Moon Day of Month", HP-65 Users' Library Program 00908A

User Instructions

NEW MOON AND FULL MOON DAY OF MONTH FAL-50

1 2

New Moon	Full Moon			P?

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	Load Side 1 and Side 2				
2	Load Data Tape (or)	.0925102612	STO	0	
		.2813291501	STO	1	
		.1703190420	STO	2	
		.06220723	STO	3	
		.11130002	STO	4	
		.02040406	STO	5	
		.07080910	STO	6	
			f	P⇄S	
		.301020	STO	0	
		.112101	STO	1	
		.220212	STO	2	
		.031323	STO	3	
		.142404	STO	4	
		.250515	STO	5	
		.061626	STO	6	
		.172707	STO	7	
		.280818	STO	8	
		.091929	STO	9	
			f	P⇄S	
3	Select Print Option 1.00 yes/o.00 no		E		1.00/0.00
4	For Date of New Moon, input month and year	mm.yyyy	A		Date_N
5	For Date of Full Moon, input month and year	mm.yyyy	B		Date_F

97 Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBLB	21 12	
002	SF2	16 21 02	Full Moon
003	*LBLA	21 11	New Moon
004	DSP4	-63 04	
005	F0?	16 23 00	
006	PRTX	-14	
007	INT	16 34	Print and Store Input
008	STOA	35 11	
009	LSTX	16-63	
010	FRC	16 44	
011	STOC	35 13	
012	EEX	-23	
013	4	04	
014	×	-35	
015	STOB	35 12	
016	2	02	Check for Jan or Feb
017	RCLA	36 11	
018	X≤Y?	16-35	
019	GSB0	23 00	
020	RCLB	36 12	
021	*LBL1	21 01	
022	0	00	
023	STO7	35 07	Check for year before 1582
024	1	01	
025	5	05	
026	8	08	
027	2	02	
028	RCLB	36 12	
029	X≤Y?	16-35	
030	GTO3	22 03	
031	1	01	
032	3	03	
033	STO7	35 07	Check for Year after 1582
034	1	01	
035	9	09	
036	0	00	
037	0	00	
038	STOD	35 14	
039	*LBL2	21 02	
040	RCLB	36 12	
041	RCLD	36 14	
042	X≤Y?	16-35	
043	GTO3	22 03	
044	EEX	-23	
045	2	02	
046	-	-45	
047	STOD	35 14	
048	1	01	
049	ST-7	35-45 07	
050	GTO2	22 02	
051	*LBL3	21 03	
052	RCLB	36 12	Calculate Julian Year
053	4	04	
054	7	07	
055	1	01	
056	2	02	
057	+	-55	
058	7	07	
059	6	06	
060	÷	-24	
061	INT	16 34	
062	STOD	35 14	
063	LSTX	16-63	
064	FRC	16 44	
065	7	07	
066	6	06	
067	×	-35	
068	RND	16 24	
069	4	04	
070	÷	-24	
071	INT	16 34	
072	STOE	35 15	← Entry Point to Table R
073	LSTX	16-63	
074	FRC	16 44	
075	4	04	
076	×	-35	
077	STO8	35 08	
078	RCLD	36 14	← Difference in Table L
079	4	04	
080	÷	-24	
081	INT	16 34	
082	ST-7	35-45 07	
083	RCLE	36 15	
084	5	05	
085	÷	-24	
086	STOI	35 46	
087	FRC	16 44	
088	1	01	
089	0	00	
090	×	-35	
091	STO9	35 09	
092	GSB9	23 09	
093	STOE	35 15	← Number from Table R
094	1	01	
095	0	00	
096	÷	-24	
097	FRC	16 44	
098	1	01	
099	0	00	
100	×	-35	
101	LSTX	16-63	
102	+	-55	
103	STOI	35 46	
104	0	00	
105	STO9	35 09	Search for Number from Table R in Table L
106	GSB9	23 09	
107	RCLE	36 15	
108	X=Y?	16-33	
109	GTO8	22 08	
110	2	02	
111	ST+9	35-55 09	
112	GSB9	23 09	

REGISTERS

0 Table R	1 Table R	2 Table R	3 Table R	4 Table M	5 Table M	6 Table M	7 Days	8 L No.	9 Control
S0 Table L	S1 Table L	S2 Table L	S3 Table L	S4 Table L	S5 Table L	S6 Table L	S7 Table L	S8 Table L	S9 Table L

A Month	B Y or Y—1	C .yyyy	D 1900 or less	E R No.	I Control

97 Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
113	RCLE	36 15	
114	X=Y?	16-33	
115	GTO8	22 08	
116	2	02	
117	ST+9	35-55 09	
118	GSB9	23 09	
119	*LBL8	21 08	
120	RCLI	36 46	
121	RCL8	36 08	Count down in table L by difference from step 77
122	-	-45	
123	1	01	
124	0	00	
125	X>Y?	16-34	
126	GTO6	22 06	
127	R↓	-31	
128	STOI	35 46	
129	*LBL7	21 07	
130	GSB9	23 09	
131	ST+7	35-55 07	← Number from Table L
132	RCLA	36 11	
133	1	01	
134	-	-45	
135	4	04	
136	÷	-24	
137	4	04	
138	+	-55	
139	STOI	35 46	
140	FRC	16 44	
141	8	08	
142	×	-35	
143	STO9	35 09	
144	GSB9	23 09	Number from Table M
145	ST-7	35-45 07	
146	F2?	16 23 02	Check for Full Moon inquiry
147	GSBe	23 16 15	
148	RCL7	36 07	
149	3	03	Check date for range between 0 and 30
150	0	00	
151	X≤Y?	16-35	
152	GSB5	23 05	
153	0	00	
154	RCL7	36 07	
155	X>Y?	16-34	
156	GTO4	22 04	
157	3	03	
158	0	00	
159	RCL7	36 07	
160	+	-55	
161	GTO4	22 04	
162	*LBL5	21 05	
163	-	-45	
164	RTN	24	
165	*LBL0	21 00	
166	RCLB	36 12	
167	1	01	
168	-	-45	

STEP	KEY ENTRY	KEY CODE	COMMENTS
169	STOB	35 12	
170	RTN	24	
171	*LBL9	21 09	
172	RCL9	36 09	
173	10ˣ	16 33	Obtain appropriate Number from selected register
174	RCLi	36 45	
175	×	-35	
176	FRC	16 44	
177	EEX	-23	
178	2	02	
179	×	-35	
180	INT	16 34	
181	RTN	24	
182	*LBL6	21 06	
183	+	-55	
184	STOI	35 46	Check for return to start of table L
185	5	05	
186	RCL9	36 09	
187	2	02	
188	+	-55	
189	X>Y?	16-34	
190	0	00	
191	STO9	35 09	
192	GTO7	22 07	
193	*LBL4	21 04	
194	EEX	-23	Complete Calculations for day of month into mm.ddyyyy display
195	2	02	
196	÷	-24	
197	RCLC	36 13	
198	LSTX	16-63	
199	÷	-24	
200	+	-55	
201	RCLA	36 11	
202	+	-55	
203	DSP6	-63 06	
204	F0?	16 23 00	
205	PRTX	-14	
206	RTN	24	
207	*LBLE	21 15	
208	F0?	16 23 00	Print Option
209	GTOa	22 16 11	
210	SF0	16 21 00	
211	1	01	
212	RTN	24	
213	*LBLa	21 16 11	
214	CF0	16 22 00	
215	CLX	-51	
216	RTN	24	
217	*LBLe	21 16 15	
218	1	01	Add for full Moon Inquiry
219	5	05	
220	ST+7	35-55 07	
221	RTN	24	

LABELS					FLAGS
A New	B Full	C	D	E Print	0 Print
a No Print	b	c	d	e Full Moon	1 ——
0 Y —1	1 Determine y	2 Repeat	3 Calc	4 Display	2 Full
5 —30	6 Move to next Col	7 Calc Day	8 Table L	9 Select No.	3 ——

SET STATUS

FLAGS	ON	OFF	TRIG		DISP	
0	□	☒	DEG	☒	FIX	☒
1	□	☒	GRAD	□	SCI	□
2	□	☒	RAD	□	ENG	□
3	□	☒			n	4

NOTES

NOTES

Hewlett-Packard Software

In terms of power and flexibility, the problem-solving potential of the Hewlett-Packard line of fully programmable calculators is nearly limitless. And in order to see the practical side of this potential, we have several different types of software to help save you time and programming effort. Every one of our software solutions has been carefully selected to effectively increase your problem-solving potential. Chances are, we already have the solutions you're looking for.

Application Pacs

To increase the versatility of your fully programmable Hewlett-Packard calculator, HP has an extensive library of "Application Pacs". These programs transform your HP-67 and HP-97 into specialized calculators in seconds. Each program in a pac is fully documented with commented program listing, allowing the adoption of programming techniques useful to each application area. The pacs contain 20 or more programs in the form of prerecorded cards, a detailed manual, and a program card holder. Every Application Pac has been designed to extend the capabilities of our fully programmable models to increase your problem-solving potential.

You can choose from:

Statistics
Mathematics
Electrical Engineering
Business Decisions
Clinical Lab and Nuclear Medicine

Mechanical Engineering
Surveying
Civil Engineering
Navigation

Users' Library

The main objective of our Users' Library is dedicated to making selected program solutions contributed by our HP-67 and HP-97 users available to you. By subscribing to our Users' Library, you'll have at your fingertips, literally hundreds of different programs. No longer will you have to: research the application; program the solution; debug the program; or complete the documentation. Simply key your program to obtain your solution. In addition, programs from the library may be used as a source of programming techniques in your application area.

A one-year subscription to the Library costs $9.00. You receive: a catalog of contributed programs; catalog updates; and coupons for three programs of your choice (a $9.00 value).

Users' Library Solutions Books

Hewlett-Packard recently added a unique problem-solving contribution to its existing software line. The new series of software solutions are a collection of programs provided by our programmable calculator users. Hewlett-Packard has currently accepted over 6,000 programs for our Users' Libraries. The best of these programs have been compiled into 40 Library Solutions Books covering 39 application areas (including two game books).

Each of the Books, containing up to 15 programs without cards, is priced at $10.00, a savings of up to $35.00 over single copy cost.

The Users' Library Solutions Books will compliment our other applications of software and provide you with a valuable new tool for program solutions.

Options/Technical Stock Analysis
Portfolio Management/Bonds & Notes
Real Estate Investment
Taxes
Home Construction Estimating
Marketing/Sales
Home Management
Small Business
Antennas
Butterworth and Chebyshev Filters
Thermal and Transport Sciences
EE (Lab)
Industrial Engineering
Aeronautical Engineering
Control Systems
Beams and Columns
High-Level Math
Test Statistics
Geometry
Reliability/QA

Medical Practitioner
Anesthesia
Cardiac
Pulmonary
Chemistry
Optics
Physics
Earth Sciences
Energy Conservation
Space Science
Biology
Games
Games of Chance
Aircraft Operation
Avigation
Calendars
Photo Dark Room
COGO-Surveying
Astrology
Forestry

CALENDARS

Everything you wanted to know about the calendar, well almost everything! If you thought that February 29th coming every fourth year was the only thing to remember about the calendar, these programs will change that opinion.

CALENDAR DATE/JULIAN DATA CONVERSION

DAYS TO DATES AND DATES TO DAYS; DAY OF WEEK

DAY OF YEAR - DAY OF WEEK

NUMBER OF WEEKDAYS BETWEEN TWO DATES

IN WHAT YEAR IS A GIVEN DATE AN M-DAY?

NUMBER OF M-DAYS BETWEEN TWO DATES AND N-TH M-DAY OF THE MONTH

HOLIDAYS

EASTER - ASH WEDNESDAY - RELIGIOUS HOLIDAYS

COMPLETE MAYA CALENDAR

MOHAMMEDAN (ISLAM) - GREGORIAN CALENDAR CONVERSION

CHINESE YEARS TO/FROM GREGORIAN YEARS

BIORHYTHM - BIOLOGICAL CYCLES

NEW MOON AND FULL MOON DAY OF MONTH

HEWLETT hp PACKARD

Reorder No. 00097-14024 Printed in U.S.A. 00097-90199