

DATE CALCULATION PROGRAMS

For SwissMicros DM32 Calculator

Based on HP-32SII

Version 1.0

Developed by

jacoher-calc

CONTENTS

1. Introduction
2. Theory of Operations
3. Variable Definitions
4. Program Descriptions
5. Operating Instructions
6. Example Calculations
7. Quick Reference
8. Program Listing

1. INTRODUCTION

This program suite provides comprehensive date calculation capabilities for the SwissMicros DM32 calculator, replicating and extending the date functions found on the classic HP-12C financial calculator. The programs enable date arithmetic, day-of-week determination, and liturgical calendar calculations.

1.1 Purpose

The program enables users to:

- Calculate days between two dates
- Add or subtract days from a given date
- Determine day of week for any date
- Calculate Easter Sunday for any year
- Calculate major Catholic moveable feast dates

1.2 Date Format

All dates use the format **M.DDYYYY** where:

- M = Month (1-12)
- DD = Day (01-31)
- YYYY = Year (four digits)

Examples: March 15, 2024 = 3.152024 | December 25, 2025 = 12.252025

1.3 System Requirements

SwissMicros DM32 calculator with RPN operating system. Program requires approximately 437 program steps.

Note: When the statefile is loaded, Flag 1 (F1) is set by default, enabling automatic day-of-week calculation with LBL D. Clear Flag 1 with [CF] [1] if this feature is not desired.

2. THEORY OF OPERATIONS

2.1 Julian Day Number (JDN)

The core of all date calculations is the Julian Day Number system, a continuous count of days since January 1, 4713 BCE (proleptic Julian calendar). By converting calendar dates to JDN, date arithmetic becomes simple integer addition and subtraction.

The programs use the proleptic Gregorian calendar algorithm with a reference offset of 1721119 (representing March 1, 0000 in the computational calendar). This approach correctly handles:

- Leap years (including century exceptions)
- Variable month lengths
- Calendar transitions across millennia

2.2 Date-to-JDN Conversion Algorithm

The conversion treats January and February as months 13 and 14 of the previous year to simplify leap year handling. The algorithm applies corrections for:

- 400-year cycle (146097 days)
- 4-year cycle (1461 days)
- Month-day offset within year

2.3 Easter Computus

Easter calculation uses the Meeus/Jones/Butcher algorithm, an optimized method for determining Easter Sunday under the Gregorian calendar rules. The algorithm accounts for:

- Golden Number (19-year Metonic cycle)
- Solar and lunar corrections
- Century adjustments
- Leap year effects

Easter is defined as the first Sunday following the first ecclesiastical full moon occurring on or after the vernal equinox (nominally March 21).

2.4 Modulo Operation

Since the HP-32SII and DM32 lack a native modulo function, the program implements $\text{MOD}(P, I)$ as: $P - (IP(I) \times I)$. This operation is fundamental to calendar calculations and day-of-week determination.

3. VARIABLE DEFINITIONS

Variable	Description
X, Y, Z, T	Original date storage (preserves input during conversions)
I	Modulo divisor (temporary in MOD function)
P	Modulo dividend (temporary in MOD function)
D	Date temporary storage (day-of-week routine)
W	Day of week result (0-6: Sun-Sat, or 1-7)
E	Easter Sunday date (M.DDYYYY format)
H	Ash Wednesday date (Easter - 46 days)
N	Ascension Thursday date (Easter + 49 days)
C	Corpus Christi date (Easter + 60 days)

Note: Registers X, Y, Z, T store the original date input (and year for Easter functions) to preserve values during internal conversions between calendar date and Julian Day Number formats.

4. PROGRAM DESCRIPTIONS

4.1 User Programs

These are the primary functions called directly by users:

LBL N - Days Between Dates (Δ DYS)

Calculates the number of days between two dates. Equivalent to HP-12C Δ DYS function.

Inputs: Date1 (Y register), Date2 (X register)

Output: Number of days (signed integer)

Operation: Converts both dates to JDN and returns the difference.

LBL D - Date Arithmetic

Adds or subtracts days from a date. Equivalent to HP-12C DATE function.

Inputs: Date (Y register), Days (X register, \pm for backward)

Output: New date in M.DDYYYY format

Flag F1: When set, also returns day of week in register W.

LBL E - Easter Date Calculator

Calculates Easter Sunday for a given year using the Meeus/Jones/Butcher algorithm.

Input: Year (YYYY format, e.g., 2024)

Output: Easter date in M.DDYYYY format (stored in register E)

LBL C - Catholic Holy Days Calculator

Calculates four major Catholic moveable feast dates based on Easter.

Input: Year (YYYY format)

Outputs (stack, bottom to top):

X: Ash Wednesday (Easter - 46 days)

Y: Easter Sunday

Z: Ascension Thursday (Easter + 49 days)

T: Corpus Christi (Easter + 60 days)

Registers: H=Ash Wed, E=Easter, N=Ascension, C=Corpus Christi

LBL B - Days from Reference Date

Calculates days between input date and a customizable reference date. The Apollo 11 moon landing (July 20, 1969 = 7.201969) is set by default as an example, but this should be personalized to any significant date: your birthday, an anniversary, an important event, or countdown date.

Input: Date (M.DDYYYYY format)

Output: Number of days from reference date

Customization: Edit line 2 of LBL B to change reference date (default: 7.201969)

4.2 Internal Subroutines

These functions are called internally and should not be executed directly by users:

LBL J: Date to Julian Day Number converter

LBL G: Julian Day Number to date converter

LBL M: Modulo (remainder) function helper

LBL O: Easter computus core algorithm

LBL W: Day of week calculator (when F1 is set)

LBL K, H, P, Q, R, S, T: Internal branching labels

5. OPERATING INSTRUCTIONS

5.1 Days Between Two Dates (LBL N)

Purpose: Calculate number of days between two dates.

Procedure:

1. Enter first date (M.DDYYYY)
2. Press [ENTER]
3. Enter second date (M.DDYYYY)
4. Press [XEQ] [N]

Result: Number of days displayed (positive if date2 > date1)

5.2 Date Arithmetic (LBL D)

Purpose: Add or subtract days from a date.

Procedure (Forward):

1. Enter date (M.DDYYYY)
2. Press [ENTER]
3. Enter number of days to add
4. Press [XEQ] [D]

Procedure (Backward):

1. Enter date (M.DDYYYY)
2. Press [ENTER]
3. Enter number of days
4. Press [+/-] to make negative
5. Press [XEQ] [D]

Optional - Day of Week:

Flag 1 is SET BY DEFAULT when the statefile is loaded, so LBL D automatically calculates day of week and stores it in register W.

- Day of week: 0=Sunday through 6=Saturday (or 1=Sunday through 7=Saturday)
- View with [RCL] [W]
- To disable this feature: [CF] [1]
- To re-enable: [SF] [1]

5.3 Easter Date (LBL E)

Purpose: Calculate Easter Sunday for any year.

Procedure:

1. Enter year (e.g., 2024)
2. Press [XEQ] [E]

Result: Easter date in M.DDYYYY format (also stored in register E)

5.4 Catholic Holy Days (LBL C)

Purpose: Calculate major moveable feast dates for a year.

Procedure:

1. Enter year (e.g., 2025)
2. Press [XEQ] [C]

Results: Four dates displayed on stack:

- Press [R↓] repeatedly to view each date
- Or use [RCL] [H], [RCL] [E], [RCL] [N], [RCL] [C]

5.5 Days from Reference Date (LBL B)

Purpose: Calculate days from Apollo 11 moon landing (July 20, 1969).

Procedure:

1. Enter date (M.DDYYYY)
2. Press [XEQ] [B]

Result: Number of days from 7.201969

To Customize Reference Date:

Edit LBL B, line 2 in program listing:

Change 7.201969 to your desired date (M.DDYYYY format)

Examples: Birthday, anniversary, countdown event, etc.

6. EXAMPLE CALCULATIONS

6.1 Days Between Dates

Problem: How many days between January 1, 2024 and July 4, 2024?

Solution:

```
1.012024 [ENTER] 7.042024 [XEQ] [N]
```

Expected Result: 185 days

6.2 Date Arithmetic - Forward

Problem: What date is 90 days after March 15, 2024?

Solution:

```
3.152024 [ENTER] 90 [XEQ] [D]
```

Expected Result: 6.132024 (June 13, 2024)

6.3 Date Arithmetic - Backward

Problem: What date was 45 days before December 25, 2024?

Solution:

```
12.252024 [ENTER] 45 [+/-] [XEQ] [D]
```

Expected Result: 11.102024 (November 10, 2024)

6.4 Date with Day of Week

Problem: What day of the week is 100 days after February 14, 2024?

Solution:

```
(Flag 1 is already set by default in the statefile)
```

```
2.142024 [ENTER] 100 [XEQ] [D]
```

```
[RCL] [W] (View day of week)
```

Expected Results:

```
Date: 5.242024 (May 24, 2024)
```

```
Day of Week: 5 (Friday; 0=Sunday or 1=Sunday depending on algorithm)
```

Note: To disable day-of-week calculation, use [CF] [1]

6.5 Easter Sunday

Problem: When is Easter Sunday in 2025?

Solution:

```
2025 [XEQ] [E]
```

Expected Result: 4.202025 (April 20, 2025)

6.6 Easter Sunday - Historical

Problem: When was Easter in 2000?

Solution:

```
2000 [XEQ] [E]
```

Expected Result: 4.232000 (April 23, 2000)

6.7 Catholic Holy Days

Problem: Calculate all major moveable feasts for 2024.

Solution:

```
2024 [XEQ] [C]
```

Expected Results (displayed on stack, X at bottom):

```
X: Ash Wednesday (H): 2.142024 (February 14, 2024)
```

```
Y: Easter Sunday (E): 3.312024 (March 31, 2024)
```

```
Z: Ascension Thursday (N): 5.092024 (May 9, 2024)
```

```
T: Corpus Christi (C): 5.302024 (May 30, 2024)
```

Use [R↓] to cycle through stack or [RCL] [H/E/N/C] to view individual dates.

6.8 Leap Year Testing

Problem: Verify leap year handling by calculating days in February 2024.

Solution:

```
2.012024 [ENTER] 3.012024 [XEQ] [N]
```

Expected Result: 29 days (2024 is a leap year)

6.9 Century Year Testing

Problem: Verify century exception - was 2000 a leap year?

Solution:

```
2.012000 [ENTER] 3.012000 [XEQ] [N]
```

Expected Result: 29 days (2000 is divisible by 400, so is a leap year)

6.10 Days from Reference Date

Problem: How many days between the moon landing (July 20, 1969) and today (Nov 2, 2024)?

Solution:

```
11.022024 [XEQ] [B]
```

Expected Result: 20,198 days

Customization Example:

To count days from your birthday (e.g., June 15, 1985):

1. Edit LBL B, line 2: Change 7.201969 to 6.151985
2. Save program
3. Run: 11.022024 [XEQ] [B] → Result: 14,385 days

7. QUICK REFERENCE

7.1 Function Summary

Label	Function	Inputs
N	Days Between Dates	Date1 [ENTER] Date2
D	Date ± Days	Date [ENTER] Days
E	Easter Sunday	Year
C	Catholic Holy Days	Year
B	Days from 7.201969	Date

7.2 Date Format

Format: **M.DDYYYY**

Examples:

- January 5, 2024 = 1.052024
- October 31, 2025 = 10.312025
- December 1, 2000 = 12.012000

7.3 Flag F1 - Day of Week

Flag 1 is SET BY DEFAULT when the statefile is loaded.

When Flag 1 is set, LBL D automatically calculates day of week:

- [RCL] [W] - View day of week (0-6 or 1-7)
- [CF] [1] - Clear flag to disable feature
- [SF] [1] - Set flag to re-enable feature

7.4 Register Storage

After running LBL C (Catholic Holy Days):

- [RCL] [H] - Ash Wednesday
- [RCL] [E] - Easter Sunday
- [RCL] [N] - Ascension Thursday
- [RCL] [C] - Corpus Christi

7.5 Important Notes

- Dates must be entered in M.DDYYYY format (no separators between components)
- Use [+/-] to make days negative for backward date arithmetic

- Flag 1 is SET BY DEFAULT - LBL D automatically calculates day of week (disable with [CF] [1])
- Programs handle leap years automatically per Gregorian calendar rules
- Easter calculations valid for years 1583 forward (Gregorian calendar adoption)
- Catholic holy days are calculated relative to Easter Sunday
- LBL B reference date (7.201969) is an example - customize to your own significant date

8. PROGRAM LISTING

The complete program listing is provided below in standard DM32 RPN format, organized by function labels. This listing can be manually entered or loaded from the state file.

LBL J - Date to Julian Day Number

```
LBL J
STO X
R↓
STO Y
R↓
STO Z
R↓
STO T
R↓
IP
LASTx
FP
100
×
FP
LASTx
IP
x<>y
1E4
×
R↑
2
-
x>0?
GTO K
12
+
x<>y
1
-
x<>y
LBL K
1
-
153
×
2
```



```

+
5
÷
IP
R↑
+
x<>y
ENTER
ENTER
100
÷
IP
146097
×
4
÷
IP
x<>y
100
XEQ M
1461
×
4
÷
IP
+
+
1721119
+
STO X
RCL T
RCL Z
RCL Y
R↑
RTN

```

LBL G - Julian Day Number to Date

```

LBL G
STO X
R↓
STO Y
R↓
STO Z
R↓

```

```

STO T
R↓
1721119
-
4
×
1
-
ENTER
ENTER
146097
÷
x<>y
LASTx
XEQ M
4
÷
IP
4
×
3
+
ENTER
ENTER
1461
XEQ M
x<>y
1461
÷
IP
R↑
IP
100
×
+
x<>y
4
+
4
÷
IP
5
×
3
-

```

```

ENTER
ENTER
153
XEQ M
x<>y
153
÷
IP
10
-
x<0?
GTO H
12
-
R↑
1
+
R↑
x<>y
R↓
R↓
LBL H
13
+
x<>y
5
+
5
÷
IP
R↑
1E4
÷
+
100
÷
+
STO X
RCL Y
RCL Z
RCL T
R↑
RTN

```

LBL N - Days Between Dates

```

LBL N
XEQ J
x<>y
XEQ J
-
RTN

```

LBL D - Date Arithmetic

```

LBL D
x<>y
XEQ J
+
XEQ G
FIX 6
FS? 1
XEQ W
FS? 1
x<>y
RTN

```

LBL W - Day of Week Calculator (F1 subroutine)

```

LBL W
STO D
XEQ J
1
+
7
XEQ M
x=0?
7
STO W
CLx
ENTER
ENTER
ENTER
FS? 1
RCL D
FS? 1
FIX 6
RCL W
RTN
RTN

```

LBL M - Modulo Function

```
LBL M
STO I
X<>Y
STO P
X<>Y
÷
IP
RCL I
×
RCL P
X<>Y
-
RTN
```

LBL O - Easter Computus Algorithm (internal)

```
LBL O
R↓
STO Y
R↓
STO Z
R↓
STO T
R↓
ENTER
ENTER
100
÷
IP
1
+
ENTER
ENTER
8
×
5
+
25
÷
IP
5
-
X<>Y
```

```

3
×
4
÷
IP
12
-
-
20
+
R↑
19
XEQ M
1
+
11
×
+
LASTx
11
÷
LASTx
-
x<>y
30
XEQ M
24
-
x=0?
GTO P
x<>y
x≤0?
GTO Q
x<>y
1
-
x≠0?
GTO P
1
+
LBL P
1
+
x<>y
LBL Q

```

```

R↓
20
x<>y
-
21
-
x≥0?
GTO R
30
+
LBL R
21
+
R↑
100
÷
IP
1
+
3
×
4
÷
IP
12
-
4
×
5
R↑
×
LASTx
R↓
x<>y
-
4
÷
IP
10
-
x<>y
+
LASTx
7
+

```

```

x<>y
7
XEQ M
-
59
+
x<>y
400
XEQ M
x=0?
GTO S
R↑
x<>y
R↓
100
XEQ M
x=0?
GTO T
R↑
x<>y
R↓
4
XEQ M
x≠0?
GTO T
LBL S
x<>y
1
+
x<>y
LBL T
R↑
+
x<>y
STO X
RCL T
RCL Z
RCL Y
R↑
RTN

```

LBL E - Easter Date Calculator

```

LBL E
XEQ O

```



```

LASTx
1E6
÷
1.01
+
XEQ J
+
1
-
XEQ G
FIX 6
RTN

```

LBL C - Catholic Holy Days Calculator

```

LBL C
XEQ E
STO E
-46
XEQ D
STO H
RCL E
49
XEQ D
STO N
RCL E
60
XEQ D
STO C
FIX 6
RCL H
RCL E
RCL N
RCL C
RTN

```

LBL B - Days from Reference Date

```

LBL B
7.201969
x<>y
XEQ N
RTN

```

Note: To customize LBL B reference date, change the second line (7.201969) to your desired date in M.DDYYYY format.

END OF MANUAL

Version 1.0

Date Calculation Programs
For SwissMicros DM32

Developed by jacoher-calc