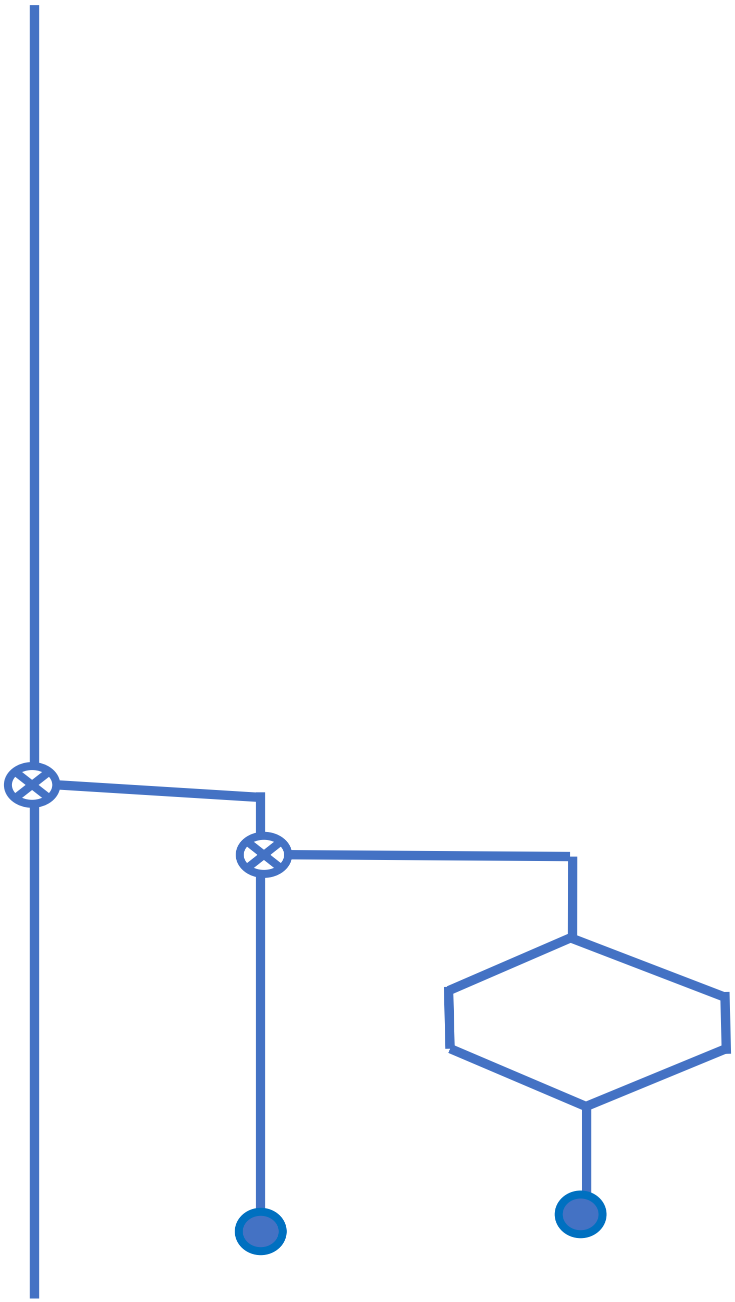


An Uber Driver Meets Forth

SVFIG

Feb. 27,2021

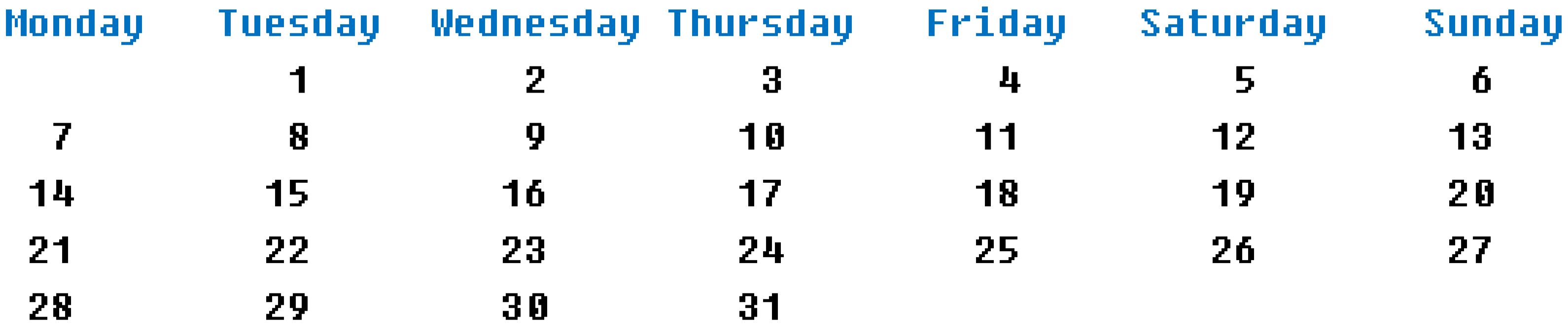
Bill Ragsdale

Our Project

An Uber driver has asked for a monthly time log for his minutes worked.

Scope

It would look something like this plus showing the minutes worked.



Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

The Driver's Input

Reset to a new month.

The Driver’s Input

Reset to a new month.

Enter today’s date.

The Driver’s Input

Reset to a new month.

Enter today’s date.

Add minutes driven after each job.

The Driver’s Input

Reset to a new month.

Enter today’s date.

Add minutes driven after each job.

Display a report and month’s summary.

Forth Input

NewMonth

TodaysDate

MinutesDriven

Report

A big caution: Error checking is needed on the operator’s input.

Log Structure

Setup an array of memory cells equivalent to a month.

Log Structure

Setup an array of memory cells equivalent to a month.

```
31 CONSTANT days/month
 7 CONSTANT days/week

CREATE DataArray days/month allot

: NewMonth DataArray days/month erase ;
```

Log Structure

Check the array allocation:

```
DataArray days/month dump

4498EC | 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

4498FC | 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00

 ok
```

A Quick Look

Here is a first cut at a month display.

```
: test1
    weeks/calendar 0
  do days/week cr  0
    do j days/week * i + cells
       DataArray + c@ 4 .r
  loop loop ;
```

A Quick Look

Here is a first cut at a month display.

```
Do a quick test of the calendar.
test1
   0   0   0   0   0   0   0
   0   0   0   0   0   0   0
   0   0   0   0   0   0   0
   0   0   0   0   0   0   0
   0   0   0   0   0   0   0
   0   0   0   0   0   0   0 ok
```

The Calendar

Let’s build a realistic calendar display.

```
: .header ."    Monday    Tuesday   Wednesday
Thursday    Friday    Saturday     Sunday" ;
```

Check the header.

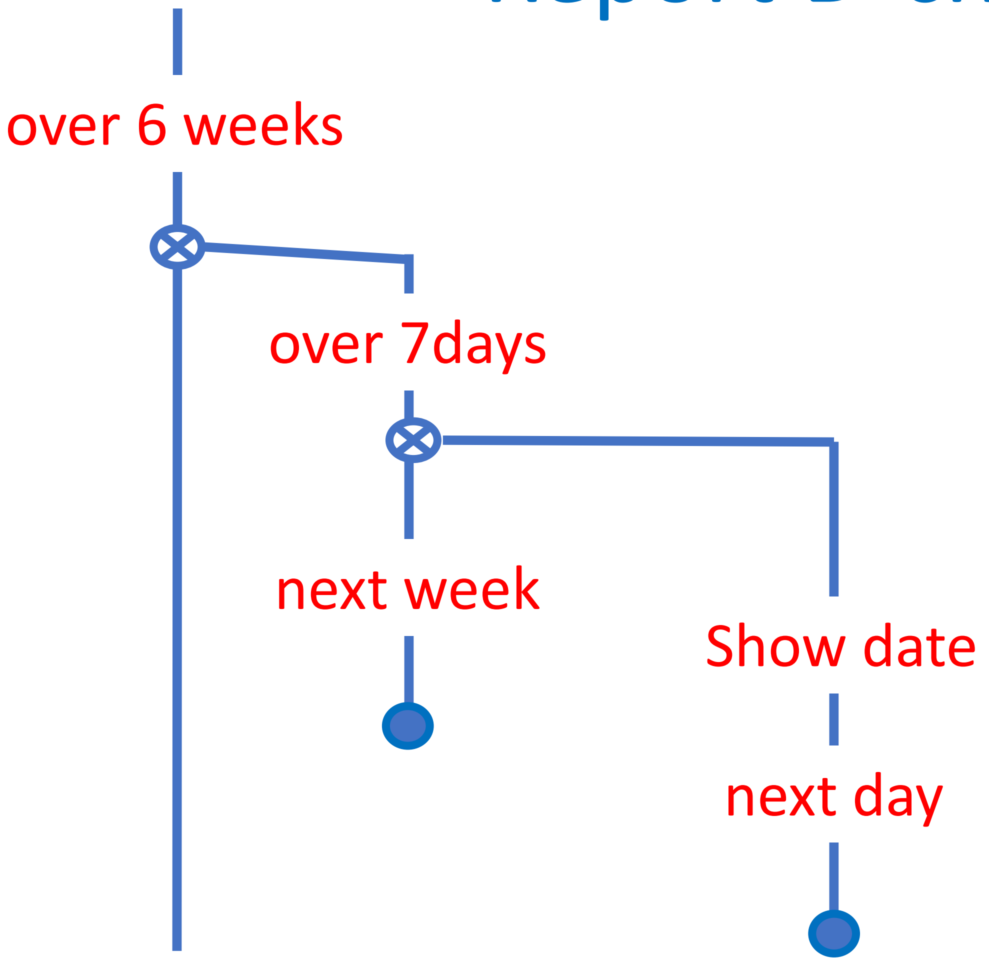
```
.header

Monday Tuesday Wednesday Thursday Friday Saturday Sunday
```

The Calendar

The realistic month display.

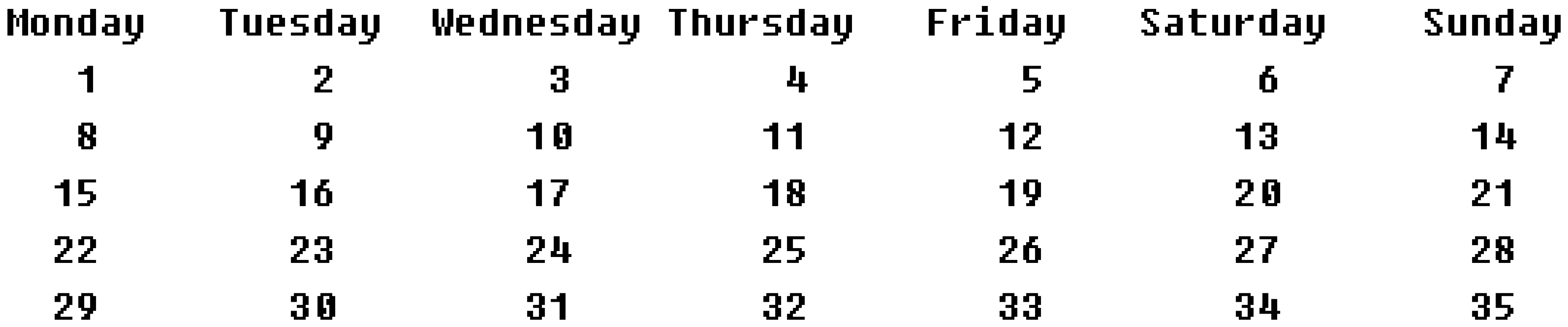
```
: Report  cr .header
  weeks/calendar  0
  do cr days/week 0
    do j days/week * i + 1+ 4 .r
       6 spaces
  loop  loop ;
```

Report D-chart



The Calendar

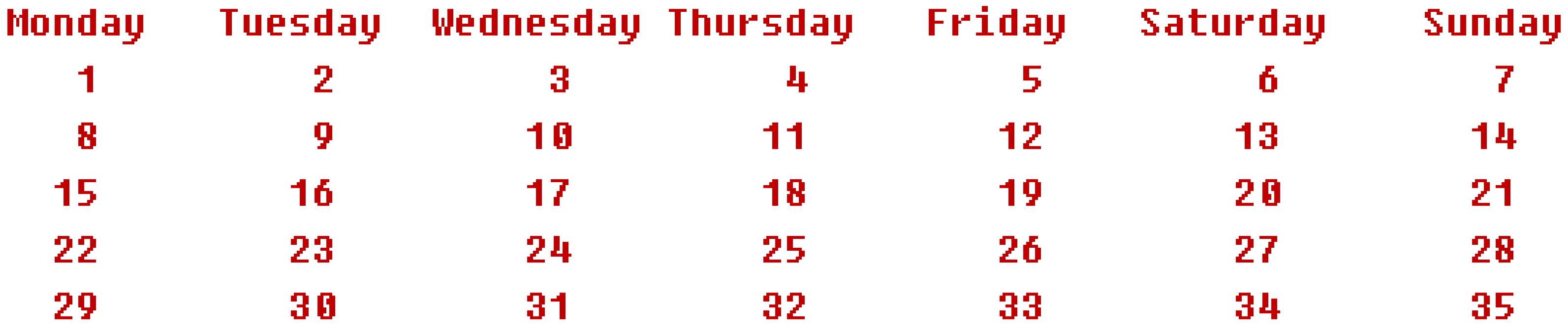
Here is a first cut at a month display.



```
Monday   Tuesday  Wednesday Thursday    Friday   Saturday    Sunday
   1         2         3          4         5         6         7
   8         9        10         11        12        13        14
  15        16        17         18        19        20        21
  22        23        24         25        26        27        28
  29        30        31         32        33        34        35
```

The Calendar

Here is a first cut at a month display.



```
Monday   Tuesday  Wednesday Thursday    Friday   Saturday    Sunday
    1         2         3         4         5         6         7
    8         9        10        11        12        13        14
   15        16        17        18        19        20        21
   22        23        24        25        26        27        28
   29        30        31        32        33        34        35
```

But. . . Too many days show.

And . . . What if the first day is not Monday?

Create An Offset

This value holds the month’s date of the first Monday. It is the first, leftmost Monday on the calendar.

```
0 VALUE FirstMonday  \ zero if Monday is first day
of the month

: IsFirstMonday  ( n --- )
     1- TO FirstMonday  ;
```

Add An Adjustment

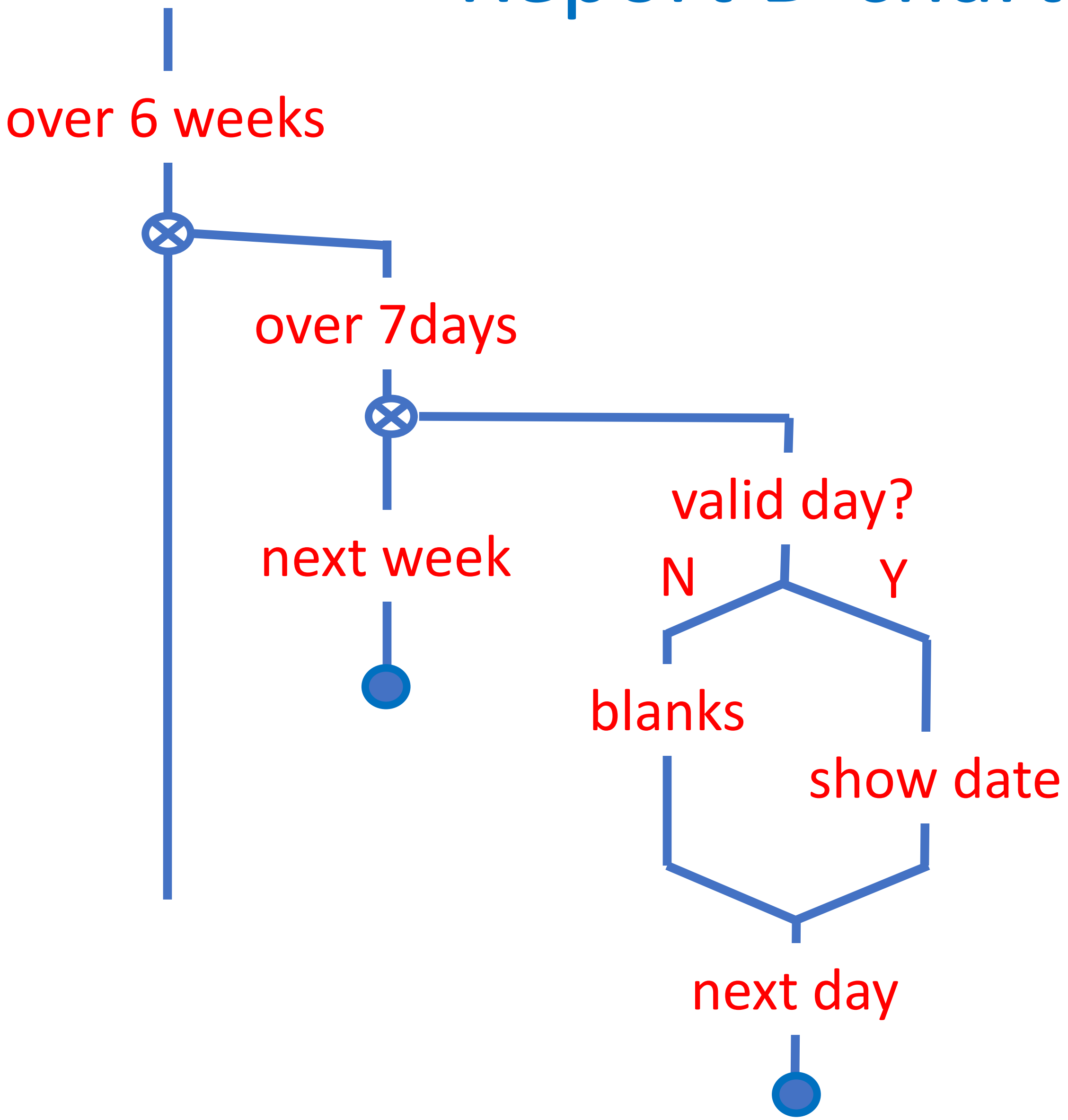
This is the offset between calendar days and the DataArray cells

```
: FirstMonday>Adjustment ( j i --- offset )
    swap days/week * +
    FirstMonday 0= if 0 else days/week
    FirstMonday - then  - ;
```

The First Monday Adjustment is a bit tricky.

```
1 1 FirstMondayAdjustment  4  ok
```

Report D-chart



over 6 weeks

over 7days

next week

valid day?

N

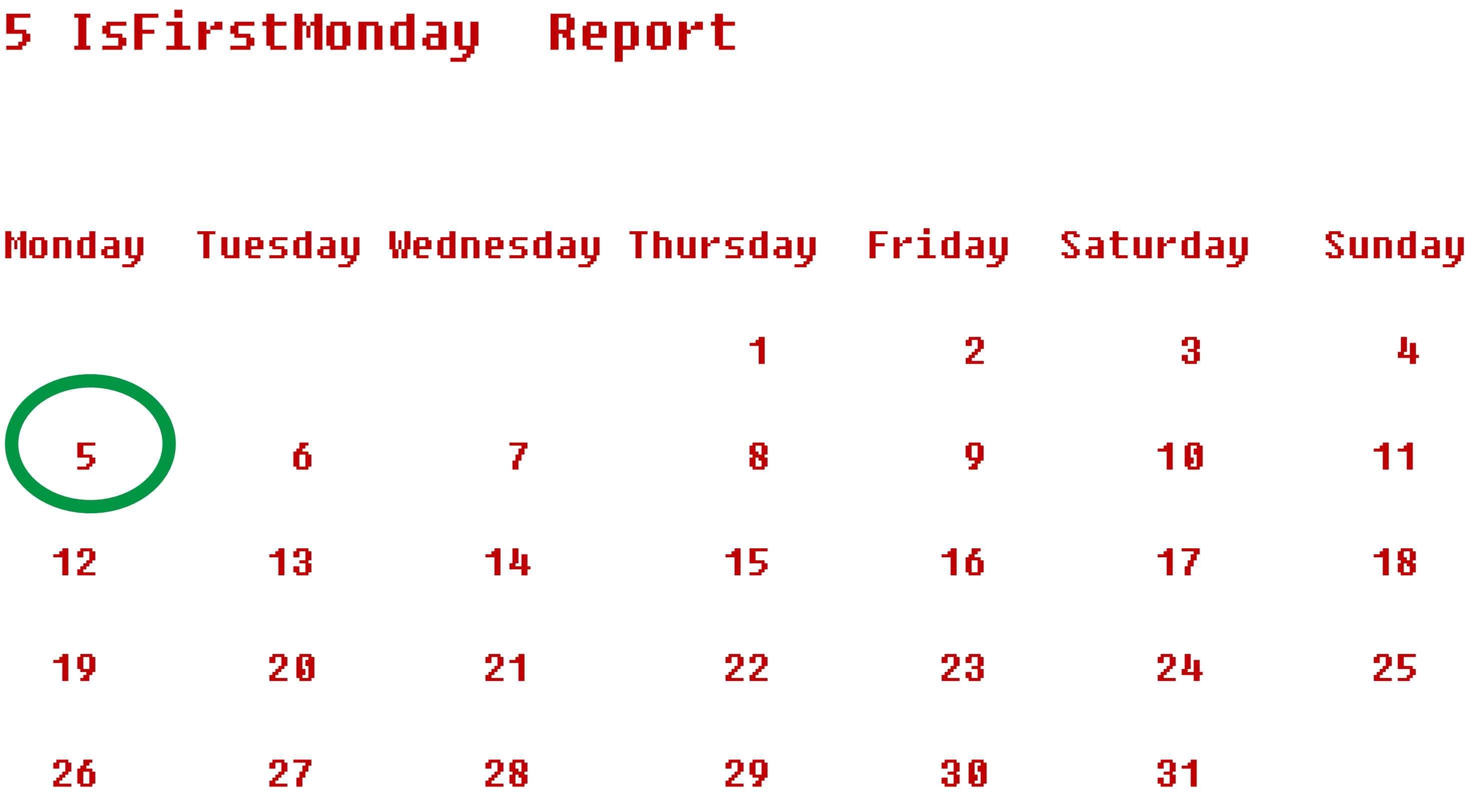
Y

blanks

show date

next day

Adjusted Calendar

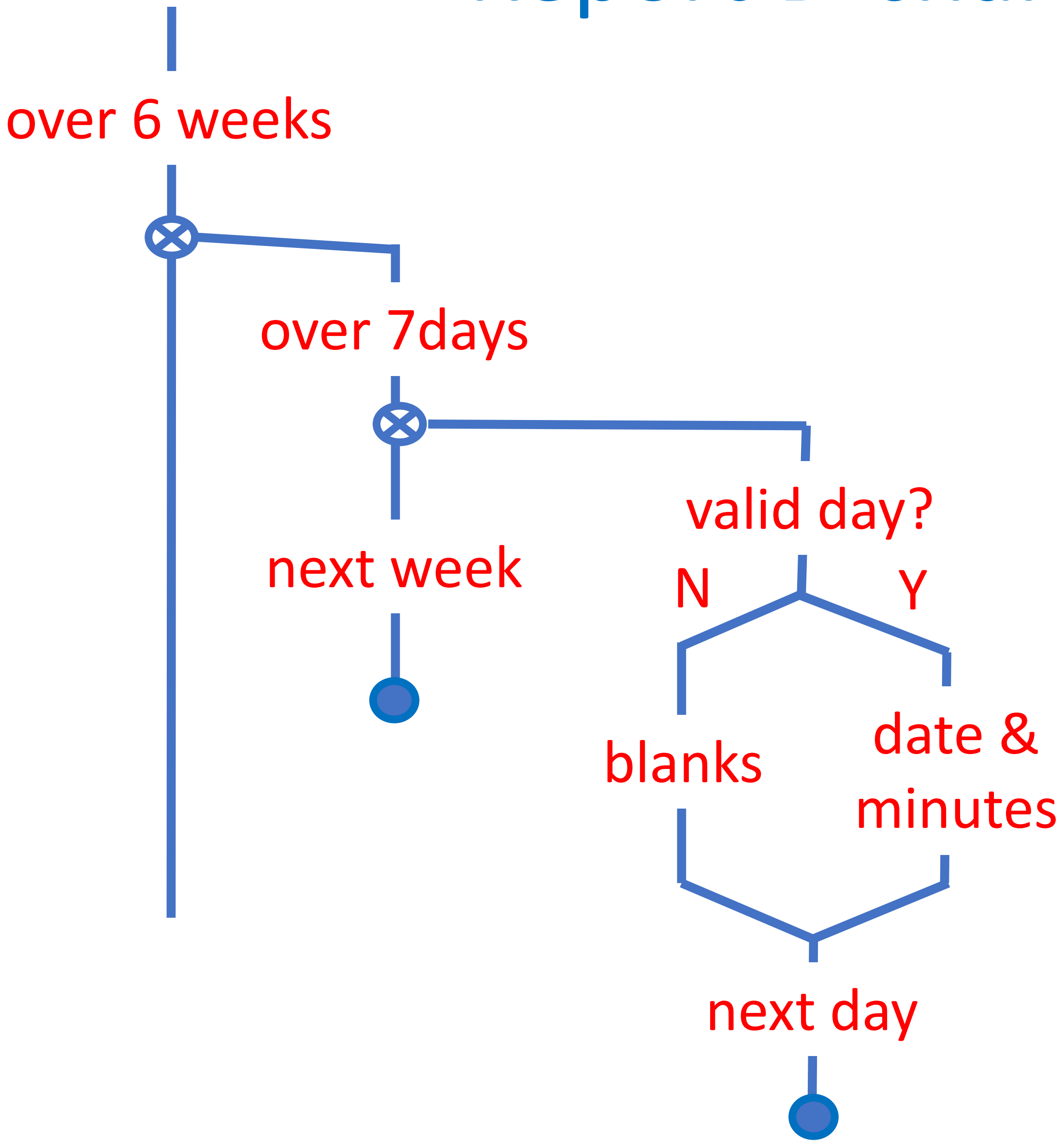


```
5 IsFirstMonday  Report


Monday   Tuesday  Wednesday Thursday  Friday   Saturday   Sunday

                               1         2         3         4
   5        6        7         8         9        10        11
  12       13       14        15        16        17        18
  19       20       21        22        23        24        25
  26       27       28        29        30        31
```

Report Pseudocode

```
Show header
over weeks
over days
  make Monday adjustment
  if  day out of range, show spaces
    else show day number
         show minutes worked for the day
    then
   loop for next day,  show new line
   loop for next week
show new line
show total minutes for the month.
```

Report D-chart



over 6 weeks

over 7days

next week

valid day?

N

Y

blanks

date & minutes

next day

Calendar II

```
: Report   cr .header
       weeks/calendar 0
    do cr   days/week 0
      do  j  i  FirstMonday>Adjustment
          dup 0< over days/month >=  or
  if ( day out of month range) drop 10 spaces
       else 1+  4 .r
         j  i  firstMonday>adjustment
         cells dataarray +  @ 5 .r  1 spaces
       then
     loop  ( for next day )
  loop   ( for next week )  cr
  Total ;
```

Calendar II

```
Monday   Tuesday  Wednesday Thursday   Friday   Saturday    Sunday

                     1   0     2   0     3   0     4   0     5   0

6   0     7   0      8   0     9   0    10   0    11   0    12   0

3   0     14   0    15   0    16   0    17   0    18   0    19   0

0   0     21   0    22   0    23   0    24   0    25   0    26   0

7   0     28   0    29   0    30   0    31   0
```

We now have 31 days . . .
An adjustment for Monday . . .
And an empty column for minutes worked.

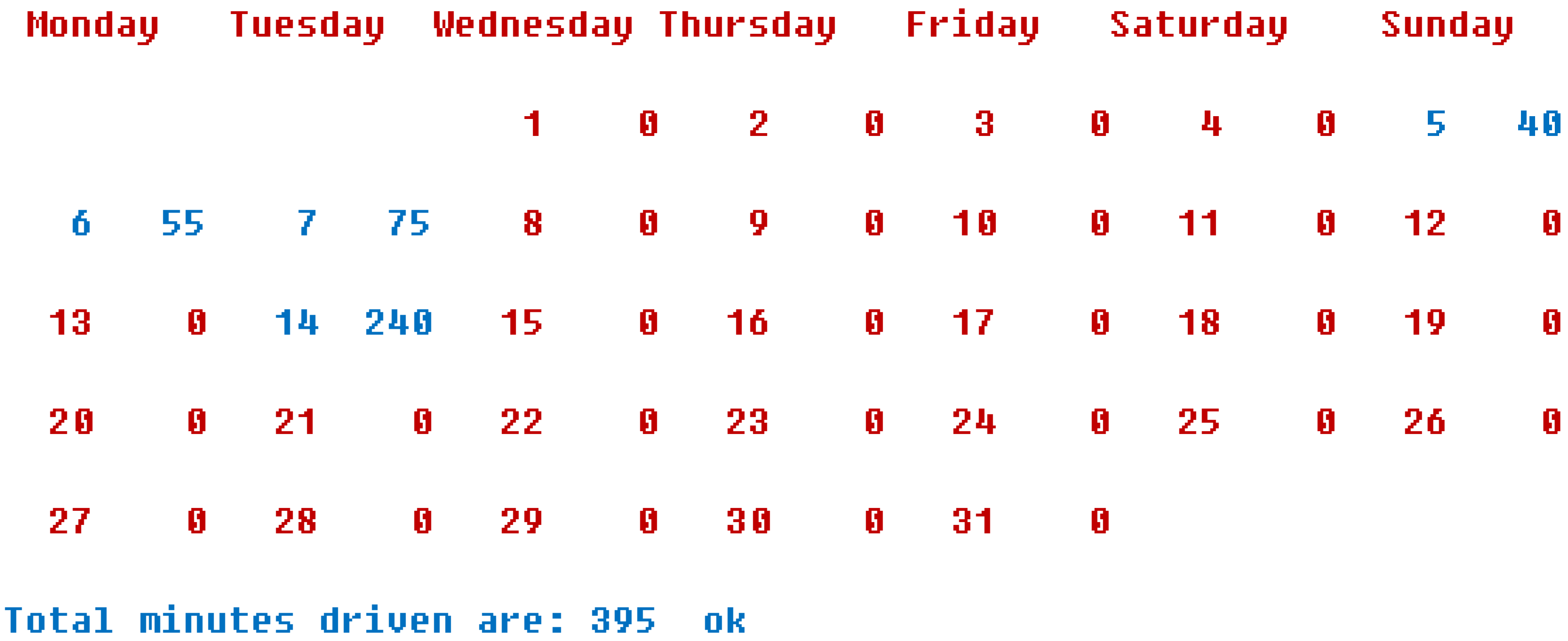
Add User Information

```
0 VALUE Today  \ day number within a month.

: NewMonth
   DataArray days/week weeks/calendar
   * cells erase ;

: TodaysDate    ( n --- )  1- TO today ;

: MinutesDriven ( n --- )
   Today Cells DataArray + +! ;
```

Sample Use, Driver Input

```
NewMonth
 6 IsFirstMonday
 5 TodaysDate    40 minutesdriven
 6 TodaysDate    20 minutesdriven
                 35 minutesdriven
 7 TodaysDate    25 minutesdriven
                 50 minutesdriven
14 TodaysDate   240 minutesdriven
Report
```

Calendar II



```
 Monday   Tuesday  Wednesday Thursday   Friday   Saturday    Sunday

                       1    0    2    0    3    0    4    0    5   40

   6   55    7   75    8    0    9    0   10    0   11    0   12    0

  13    0   14  240   15    0   16    0   17    0   18    0   19    0

  20    0   21    0   22    0   23    0   24    0   25    0   26    0

  27    0   28    0   29    0   30    0   31    0

Total minutes driven are: 395  ok
```

Future?

Add input error checks.

Future?

Add input error checks.

Printed reports.

Future?

Add input error checks.

Printed reports.

Correlate with revenue and expenses.

Future?

Add input error checks.

Printed reports.

Correlate with revenue and expenses.

Smartphone touch-screen.

Future?

Add input error checks.

Printed reports.

Correlate with revenue and expenses.

Smartphone touch-screen.

Graphical buttons and keypad.

Future?

Add input error checks.

Printed reports.

Correlate with revenue and expenses.

Smartphone touch-screen.

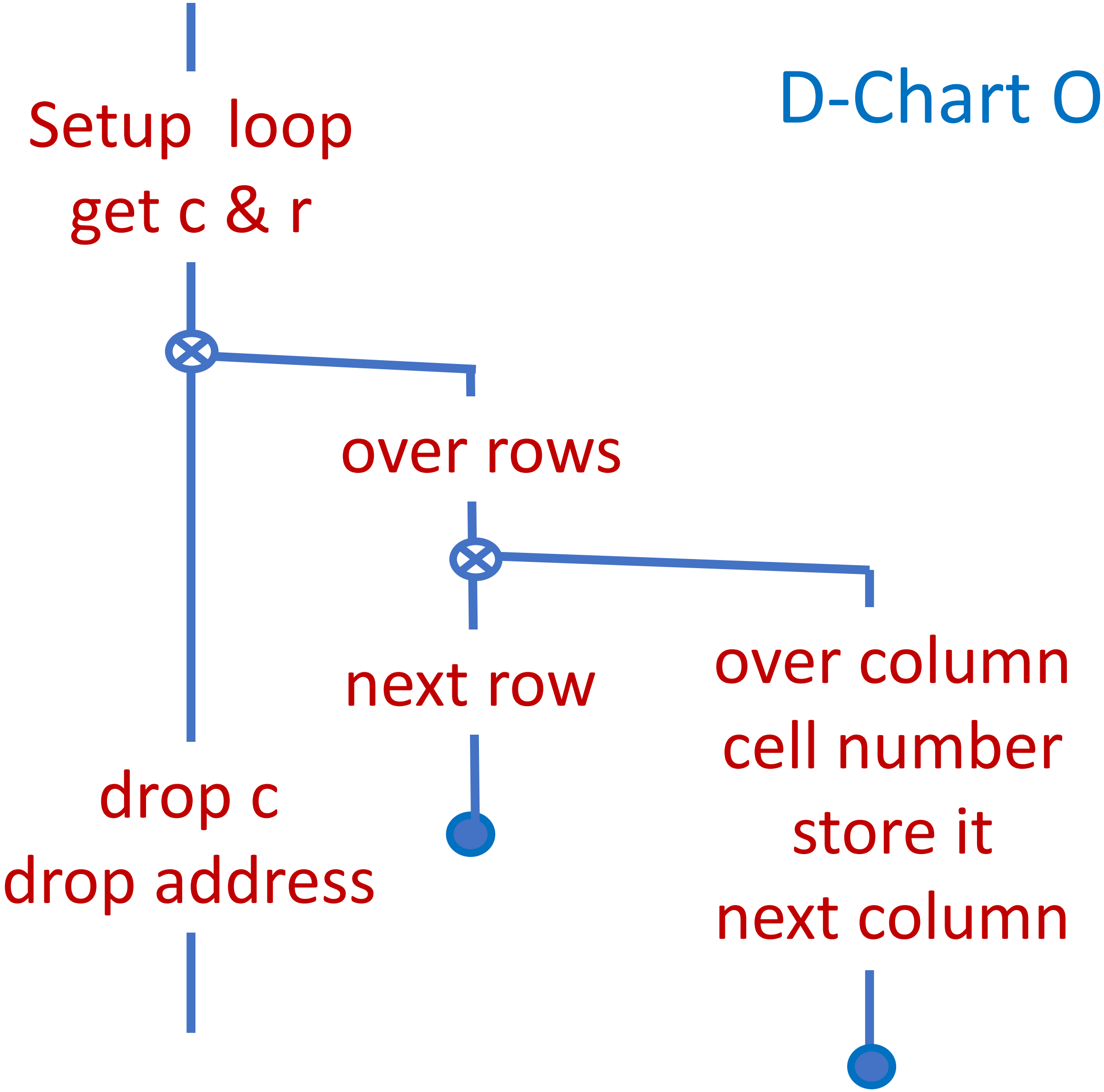
Graphical buttons and keypad.

Spinners instead of keypad.

References

- https://github.com/BillRagsdale/Forth_Projects
- File: UberD.f

D-Chart Of }fill



Setup loop
get c & r

over rows

next row

over column
cell number
store it
next column

drop c
drop address

Summary

I use MatLab and the clone Octave.

I have a project about once a year and need a complete refresher on the semantics.

The manuals explain each single structure but not the overall parsing order.

Therefor I need a reverse Polish syntax that needs no ‘refresher course’.

Matrix Forth is the answer.

Add An Adjustment

```
0 VALUE FirstMonday

: IsFirstMonday  ( n --- )
     1- TO FirstMonday  ;
```

Enter and check the First Monday value. See 3:

```
4 IsFirstMOnday FirstMonday

3  ok
```