♥MoHPC♠ *The Museum of HP Calculators*

HP Forum Archive 19

[Return to Index | Top of Index]

HP-15C Mini-Challenge: Magic squares !

*Message #1 Posted by **Valentin Albillo** on 10 June 2009, 6:28 a.m.*

Hi all:

Since these next four days will be a loooong weekend here in Madrid, I feel like concocting a brand new ***HP-15C Mini-Challenge***^{TM} for you to try, namely:

A **magic square** is a square arrangement of numbers such that *each* row, column, and main diagonals add up to the *same* value, say **K**, which is called the magic square's *constant*.

Now let's say you're given an *arbitrary 3rd order magic square* which has some (not given) value **K** as its constant. You must write a routine which takes the given magic square as its input and then computes the elements of *another* magic square of the same order but having $\mathbf{K}^3$ as its constant instead.

You don't need to include any input or output procedures in your routine, it may assume that the given magic square is already stored in memory in any way you find convenient (such as sequential numeric registers, or a matrix, or whatever) and it just needs to *compute* the elements of the new magic square and leave them stored in memory, no need to actually output them to the display.

For example, if given the following magic square:

```
 71   89   17
  5   59  113
101   29   47
```

which has **K** = **177** (all rows, columns, and main diagonals add up to **177**) you must compute and store the elements of another magic square of the same order but having **K** = 177^3 = **5,545,233**.

You must optimize for program length, the shorter the better. Not counting label and return, it can be done in the number of steps corresponding to a very famous integer traditionally considered having to do with luck (whether good or bad is up to you ...), and further an additional step can be shaved off with one little proviso.

I'll provide my solution and comments next Monday or so, maybe earlier if someone stumbles upon it. Have a nice weekend and all that, and

Best regards from V.

Edited for typos

Edited: 10 June 2009, 6:39 a.m.

Re: HP-15C Mini-Challenge: Magic squares !

Message #2 Posted by ***Eamonn*** *on 10 June 2009, 11:13 a.m.,*
in response to message #1 by Valentin Albillo

Hi Valentin,

Great to see another 15c Mini challenge.

Here is my solution - 4 steps excluding the LBL and RTN. The solution assumes that the magic square is stored in a Matrix and that the Matrix descriptor is in the X-register before being called.

```
LBL A
ENTER
f MATRIX 7
x^2
*
RTN
```

It could be reduced to 3 steps if the matrix descriptor is in both the X and Y registers before the routine is called. In this case the first ENTER can be removed.

The only caveat is that it does not work for magic squares which have negative entries for some of the elements.

Best regards,

Eamonn.

Re: HP-15C Mini-Challenge: Magic squares !

Message #3 Posted by ***Didier Lachieze*** *on 10 June 2009, 12:45 p.m.,*
in response to message #2 by Eamonn

Great to see a mini challenge I can solve !! :-)

My solution, assuming that the magic square is in matrix A: 7 steps excluding LBL and RTN, should work for all magic squares.

```
01 LBL A
02 f RESULT B
03 RCL MATRIX A
04 ENTER
05 *
06 f RESULT C
07 RCL MATRIX A
08 *
09 RTN
```

Result is in matrix C and is different from the one obtained with Eamonn solution.

Edited: 10 June 2009, 12:47 p.m.

Re: HP-15C Mini-Challenge: Magic squares !

Message #4 Posted by ***Eamonn*** *on 10 June 2009, 7:57 p.m.,*
in response to message #3 by Didier Lachieze

Hi Didier,

I like your solution. I was not aware that the third power of a magic square matrix is itself a magic square matrix with a magic sum that is the third power of the magic sum of the original matrix. Very nice result.

I played around with some more ideas and here is another seven step solution (not including the LBL and RTN) that I believe solves the challenge as stated and generates a solution that is different to the earlier posted solutions.

This solution requires that the original magic square be stored in the 3x3 Matrix A, with the center square stored in A(1,1). The other values from the original square can be stored in any order in A that is convenient for the user.

```
LBL A
f MATRIX 1
RCL A
3
y^x
```

```
9
*
STO MATRIX A
RTN
```

The result is returned in the Matrix A, with the very magic property that the new magic square can be read from Matrix A in conventional fashion (i.e. the center of the new magic square ends up in A(2,2), the top left value is in A(1,1), etc. This solution should work for all 3x3 magic squares, even those with a mix of positive and negative values, always returning a square in which all the rows and columns and the main diagonals sum to the cube of the magic number of the original square.

Best regards,

Eamonn.

Re: HP-15C Mini-Challenge: Magic squares !

Message #5 Posted by ***Didier Lachieze*** *on 10 June 2009, 9:53 p.m.,*
in response to message #4 by Eamonn

Well, I realize that I've been a bit quick to jump to a solution. I found it quite intuitively and I've verified it for several magic squares but I don't have a complete demonstration ...

Here are some explanations: let's take two 3x3 magic squares X, constant K and Y, constant L

```
    ( a, b, c )     ( r, s, t )
X = ( d, e, f )  Y = ( u, v, w )
    ( g, h, i )     ( x, y, z )
```

then

```
      ( ar + bu + cx, as + bv + cy, at + bw + cz )
X*Y = ( dr + eu + fx, ds + ev + fy, dt + ew + fz )
      ( gr + hu + ix, gs + hv + iy, gt + hw + iz )
```

the sum of the first column is:

```
(a+d+g)r + (b+e+h)u + (c+f+i)x = Kr + Ku +Kx = K(r+u+x) = K*L
```

and you can see in the same way (using only rows and columns constant property) that the result is the also K*L for all rows and columns.

However I've not found a way to prove it's the same for the main diagonals ... btw it's not true as on a few examples I've tried the diagonal constant is not K*L.

If we suppose that the rows and columns constant of X*Y is K*L then for X*X it's K^2 and for X*X*X it's K^3; and strangely this last one is also true for the diagonals for the different magic squares I've tried so far.

So I still have to prove that the constant for the diagonals of X*X*X is K^3...

Now, to add something else to this incomplete and potentially wrong solution, and as Valentin has expressed previously his interest in the old Sharp Pocket Computers, here is a program for the Sharp PC-1403 using the Matrix functions entry points described earlier on this forum in this thread:

The Sharp PC-1403 uses 3 matrix: X and Y for calculation, M for memory.

The program below assumes that the magic square is in X and return the result of X^3 in X.

```
10:"A":CALL 26191:REM X->M
20:CALL 26186: REM X^2->X
30:CALL 26163: REM X<>Y
40:CALL 26199: REM M->X
50:CALL 26119: REM X*Y->X
60:PRINT X(0,0)+X(1,1)+X(2,2),X(0,2)+X(1,1)+X(2,0)
```

Example:

```
CALC mode
SHIFT down                    'to enter in Matrix mode
3 ENTER 3 ENTER               'to dimension matrix X as 3x3
 71 ENTER 89 ENTER  17 ENTER 'enter first row
  5 ENTER 59 ENTER 113 ENTER 'enter second row
101 ENTER 29 ENTER  47 ENTER 'enter third row
3 ENTER 3 ENTER               'to dimension matrix Y as 3x3
ENTER (9 times)               'until you get MATRIX OPERATIONS on the display
BASIC mode
DEF A                         'to run the program
```

It ends by printing the constant for the 2 main diagonals, so you can verify it's K^3. For the example above it shows:

```
5545233.       5545233.
```

Edited to add some precisions to the explanations.

Edited: 11 June 2009, 3:25 a.m.

Re: HP-15C Mini-Challenge: Magic squares !

Message #6 Posted by ***Eamonn*** *on 12 June 2009, 1:18 p.m.,* *in response to message #5 by Didier Lachieze*

Hi Didier,

Section 8 of the book "Mathematical Explorations with MATLAB" deals with Magic squares. Part of exercise (v) on Page 104 is to prove the the product of an odd number of magic squares is also a magic square. The relevant sections are viewable in Google books (here is a link: Mathematical Explorations with MATLAB).

So, it appears that your solution is good.

Best regards,

Eamonn.

Re: HP-15C Mini-Challenge: Magic squares !

Message #7 Posted by ***Didier Lachieze*** *on 12 June 2009, 5:42 p.m.,*
in response to message #6 by Eamonn

Eamonn, thanks for the link, I will check it. I've prepared my solution below before seeing your post so I'm posting it anyway as it reflect my findings.

Introduction

To provide some formal content to my 15C program submitted above, here is a complete demonstration showing that if $\mathbf{M}$ is a 3x3 magic square having $\mathbf{K}$ as its constant, then $\mathbf{M}^3$ is also a magic square with $\mathbf{K}^3$ as its constant.
I suppose there are some smarter and nicer demonstrations as mine is a bit on the "brute force" side, being based on the direct calculation of each term of $\mathbf{M}^3$.

The trick

After several tries, I've come to the conclusion that I needed to take advantage of the magic square properties to reduce the number of variables to a minimum.

As I was manipulating 3x3 magic squares I've found an interesting property:

$$M = \begin{pmatrix} r, & s, & t \\ u, & v, & w \\ x, & y, & z \end{pmatrix}$$

being a magic square with K as it's constant, we can write using the rows properties:

$$3K = (r+s+t) + (u+v+w) + (x+y+z)$$

Rearranging the terms to use also the diagonals properties we have:

$$\begin{aligned}
3K &= (r+v+z) + (x+v+t) + (s+u+w+y) -v \\
3K &= (r+v+z) + (x+v+t) + (s+v+y -v + u+v+w -v) -v \\
3K &= (r+v+z) + (x+v+t) + (s+v+y) + (u+v+w) -3v \\
3K &= 4K -3v \\
3v &= K
\end{aligned}$$

So: **v = K/3**

This tell us one thing about 3x3 magic squares: their constant is always a multiple of 3.

Having a direct relation between the magic square constant **K** and the center value, we can now write **M** differently:
let's name **a** the central value and use it as a reference, expressing each value as **a** + or - something:

$$M = \begin{pmatrix} a+b & a-b-c & a+c \\ a-b+c & a & a+b-c \\ a-c & a+b+c & a-b \end{pmatrix}$$

Now we have an expression of **M** using only the central value **a** (which is equal to one third of the constant **K**) plus 2 others parameters **b** & **c** that can be positive or negative and are sufficient to completely define the magic square.

Calculating M^2

If we calculate **M*M**, for the first element (top left) we have:

$$\begin{array}{lll}
\ \ (a+b)(a+b) & & a^2 + ab + ab + b^2 \\
+ (a-b-c)(a-b+c) & = & + a^2 - ab + ac - ab + b^2 - bc - ac + bc -c^2 \\
+ (a+c)(a-c) & & + a^2 - ac + ac - c^2
\end{array}$$

$$= \quad 3a^2 + 2b^2 - 2c^2$$

after calculating all terms we get the following matrix:

$$M*M = \begin{pmatrix} 3a^2+2b^2-2c^2 & 3a^2-b^2+c^2 & 3a^2-b^2+c^2 \\ 3a^2-b^2+c^2 & 3a^2+2b^2-2c^2 & 3a^2-b^2+c^2 \\ 3a^2-b^2+c^2 & 3a^2-b^2+c^2 & 3a^2+2b^2-2c^2 \end{pmatrix}$$

We can see that for $\mathbf{M^2}$ each row and column as well as the diagonal bottom left to top right has the same sum which is $\mathbf{9a^2} = \mathbf{K^2}$ (remember: $\mathbf{a = K/3}$) , but the remaining diagonal (top left to bottom right) has a different sum so $\mathbf{M^2}$ is not a magic square.

Calculating M^3

Now let's calculate $\mathbf{M^2*M}$.

First, what's the center value of $\mathbf{M^3}$:

$$\begin{array}{ll} (3a^2-b^2+c^2)(a-b-c) & \quad 3a^3 - ab^2 + ac^2 - 3a^2b + b^3 + bc^2 - 3a^2c + b^2c - c^3 \\ + (3a^2+2b^2-2c^2)a & = + 3a^3 + 2ab^2 - 2ac^2 \\ + (3a^2-b^2+c^2)(a+b+c) & \quad + 3a^3 - ab^2 + ac^2 + 3a^2b - b^3 - bc^2 + 3a^2c - b^2c + c^3 \end{array}$$

$$\begin{array}{lll} = & 9a^3 & \text{(we can simplify a lot of terms above)} \\ = & 9(K/3)^3 & \\ = & \mathbf{K^3/3} & \end{array}$$

That starts to look interesting as we want to prove that $\mathbf{M^3}$ is a magic square with $\mathbf{K^3}$ as constant...

Let's continue with all terms (it took me some time to get it right ...) and we have:

$$M^3 = \begin{pmatrix} 9a^3+3b^3-3bc^2 & 9a^3-3b^3+3c^3+3bc^2-3b^2c & 9a^3-3c^3+3b^2c \\ 9a^3-3b^3-3c^3+3bc^2+3b^2c & 9a^3 & 9a^3+3b^3+3c^3-3bc^2-3b^2c \\ 9a^3+3c^3-3b^2c & 9a^3+3b^3-3c^3-3bc^2+3b^2c & 9a^3-3b^3+3bc^2 \end{pmatrix}$$

If we add up each row, column or diagonal we can see now that $\mathbf{M^3}$ is a magic square with a constant of: $27a^3 = 27(K/3)^3 = \mathbf{K^3}$, so we are done !!!

Conclusion

As a conclusion, many thanks to Valentin for crafting the Mini-challenges. This one has been for me both challenging and rewarding as I've been able to solve it practically with the requested 15C program based on my intuitions and theoretically with a formal demonstration. Overall this was an excellent brain exercise !

Edited to fix a few typos.

Edited: 12 June 2009, 7:00 p.m.

Re: HP-15C Mini-Challenge: Magic squares !

Message #8 Posted by ***Valentin Albillo*** *on 16 June 2009, 8:26 a.m.,*
in response to message #7 by Didier Lachieze

Hi, Didier:

> Thanks for your appreciation and congratulations for your correct solution and excellent further exposition.
>
> My original solution was the following 7-step HP-15C code:

```
    RCL MATRIX A
    ENTER
    ENTER
    RESULT B
     *
    RESULT C
     *
```

> which could be shortened by one step by making A one of the result matrices (instead of B or C) and taking advantage of the fact that after a memory reset the A matrix is the default result matrix, but that's too specific and contrived and would work just once, unlike the general, repeatable 7-step solution above.
>
> When applied to

```
    6  1  8
    7  5  3   K = magic constant = 15
    2  9  4
```

the above code quickly produces:

```
    1101  1221  1053
    1077  1125  1173    K2 = magic constant = K^3 = 15^3 = 3375
    1197  1029  1149
```

as required.

Just in case you're interested in trying some more of my HP-related challenges, here you are, some links to most of the ones I posted to this forum in the past years, complete with all contributions and full commented solutions:

Short & Sweet Math Challenge #20

Short & Sweet Math Challenge #19

Short & Sweet Math Challenge #18

Short & Sweet Math Challenge #17

Short & Sweet Math Challenge #16

Short & Sweet Math Challenge #15

Short & Sweet Math Challenge #14

Short & Sweet Math Challenge #13

Short & Sweet Math Challenge #12

Short & Sweet Math Challenge #11

Short & Sweet Math Challenge #10

Short & Sweet Math Challenge #9

Short & Sweet Math Challenge #8

Short & Sweet Math Challenge #7

Short & Sweet Math Challenge #6

Short & Sweet Math Challenge #5

Short & Sweet Math Challenge #4

Short & Sweet Math Challenge #3

Short & Sweet Math Challenge #2

Short & Sweet Math Challenge #1

Valentine's Day 2008 Mini-Challenge

Valentine's Day Mini-Challenges

HP-15C Mini-challenge

HP-15C Mini-challenge

HP-15C Mini-challenge

HP-15C Mini-challenge

Mini-Challenge

Arguably useful Mini-challenge

A little HP-71B challenge

HP42S Mini-challenge

A little classical RPN challenge

A new small HP-15C challenge

Really really small HP-15C Mini-challenge

New HP-15C Mini-challenge

HP-15C Mini-challenge

A little HP-15C challenge

A little HP-15C challenge

Two little RPN challenges

Again, thanks to everyone for your appreciation and your contrubutions, glad you liked it, and

Best regards from V.

Re: HP-15C Mini-Challenge: Magic squares !

Message #9 Posted by ***Egan Ford*** *on 16 June 2009, 9:55 a.m.,*
in response to message #8 by Valentin Albillo

Valentin, thanks for contriving this challenge. I look forward to many more.

Re: HP-15C Mini-Challenge: Magic squares !

Message #10 Posted by ***Didier Lachieze*** *on 16 June 2009, 12:08 p.m.,*
in response to message #8 by Valentin Albillo

Hi Valentin,

I find your original solution nicer than mine, even if they do the same thing in the same number of steps. Yours is more elegant with a similar structure for the two operations and looks simpler.

This is, I suppose, the Zen of RPN coding:

1) Operand:

```
    RCL MATRIX A
    ENTER
    ENTER
```

2) First operation:

```
    RESULT B
     *
```

2) Second operation:

```
    RESULT C
     *
```

While my solution is mixing data, operations and result assignments and at the end seems more complex and less elegant (at least to me):

```
    RESULT B
    RCL MATRIX A
    ENTER
    *
    RESULT C
    RCL MATRIX A
    *
```

... a lot could be learned from even the smallest code.

The list of your HP-related challenges is amazing ! I've not realized there were so many as I'm quite a newbie on this forum (I've been hooked again on HP calculators last year when I got a 11C and a 35s, nearly 30 years after my 41C).

Didier

Re: HP-15C Mini-Challenge: Magic squares !

Message #11 Posted by ***Marcus von Cube, Germany*** *on 16 June 2009, 1:41 p.m.,*
in response to message #8 by Valentin Albillo

Hi Valentin,

I didn't try your challenge but I've learned something by walking through this thread. Thanks for your efforts and ideas! Could you create an article with just the link list from this post. It could serve as a reference.

Marcus

Re: HP-15C Mini-Challenge: Magic squares !

Message #12 Posted by ***PeterP*** *on 16 June 2009, 4:46 p.m.,* *in response to message #11 by Marcus von Cube, Germany*

I'd like to second all of Markus' comments, an article with all the links would be most fabulous.

(I had come across that particular questions for magic squares when I was high-school and part of a little news-paper editor group with magic squares and the like where I had to come up with new magic squares on a regular basis... Without an HP calcs unfortunately, but a TI 52-II with I think 12 steps or something the like)

Thanks for your postings, its good to see you more around again!

Cheers

Peter

Re: HP-15C Mini-Challenge: Magic squares !

*Message #13 Posted by **Bruno Férard** on 21 June 2009, 8:53 a.m.,*
in response to message #8 by Valentin Albillo

It's again a real pleasure to ponder over your challenge. It was both entertaining and instructive.

Many thanks, I am looking forward to trying the next one. (I am going to bookmark the index of former challenges)

[Return to Index | Top of Index]

GTO Go back to the main exhibit hall